

**THE FINAL YEAR PROJECT**

**WXET 3182**

**ATTENDANCE  
MANAGEMENT SYSTEM**

**BY**

**GOH SZE ENG**

**WET 98065**

**Under the supervision of**

**Mr. Ling Teck Chaw**

**Session 2000/2001**

---

Submitted as a partial requirement for the  
Degree of Bachelor of Information Technology  
University of Malaya

## ABSTRACT

Attendance Management System (AMS) is developed to automate entire workflow of the attendance logging system of Faculty of Computer Science and Information Technology. This system is currently in used by all the staffs in faculty.

The main purpose of this project is to enhance the current system and add new functionalities to the system. The lacking of the system and all the problems faced by users will be identified in the review of literature. With the findings and the gathered information, all the functions in the system will be consolidated and the feasibility of each module will be tested. Calendar system is the new features that will be added to the system so as other new functionalities for the advantage and convenient of users.

Under this project, the current database will be examined to reduce the redundancy of data and to improve its performance in term of storage and processing. Some of the tables will be normalized and the relationship between these tables will be refined too. New scheduled jobs are created to meet the needs and requirements of users as well as faculty. The interface of the system is designed and changed to be more consistent and user-friendly.

Other than that, this project will also focus on the enhancement of integration with Leave Management System. The integration is carried out step by step from the development phase to implementation phase till the testing phase. Thus, the existing system will be the basis from which the better system will be developed.

By making modification and adding new functionalities to the existing system, it is hoped to make the entire process of attendance faster, precise and flawless. Finally, the completion of this system will surely bring a lot of benefits to the faculty.



## ACKNOWLEDGEMENT

First of all, I would like to extend my sincere appreciation and immense gratitude to Mr. Ling Teck Chaw, my supervisor who has given me the opportunity to enhance this system and provided me with sufficient facilities. I would like to thank him for his constructive advice, invaluable guidance, insight and encouragement throughout the entire project.

Then, my special thanks would like to go to Miss Nor Aniza, my project moderator who contributed suggestions and ideas to further enhance value of this project. My appreciation will also like to go to Deputy Dean, Dr. Diljit Singh and the head department of Artificial Intelligent, Dr. Selva who have provided the valuable information. And also not forget the system administrator, Puan Azlin, who has patiently explained to me the problems faced in this system as well as the suggestions to solve those problems.

Last but not least, I would like to thank my teammate Miss Chan Yim Khim, my senior, Mr. Ang Tan Fong and all my course mates for their valuable ideas. Only with their helps and supports, I could complete my projects successfully.

Thank you very much!

## CONTENTS

<b>ABSTRACT.....</b>	<b>I</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>II</b>
<b>CONTENTS.....</b>	<b>III</b>
<b>LIST OF FIGURES.....</b>	<b>VII</b>
<b>LIST OF TABLES.....</b>	<b>IX</b>
 <b>CHAPTER 1 : INTRODUCTION.....</b>	 <b>1</b>
1.1 Overview of Attendance Management System.....	2
1.2 Project Objectives.....	3
1.3 Project Scope.....	5
1.3.1 User Module.....	5
1.3.2 Administrator Module.....	5
1.4 Project Motivation.....	6
1.5 Project Schedule.....	7
1.6 Report Organization.....	9
 <b>CHAPTER 2 : LITERATURE REVIEW.....</b>	 <b>11</b>
2.1 Purpose.....	11
2.2 Approach.....	11
2.2.1 Review of the Existing System.....	11
2.2.1.1 The System Deficiency and Weaknesses.....	12
2.2.2 Interview.....	13
2.2.3 Internet Research.....	14
2.2.3.1 Study on Similar Attendance Management Systems.....	14
2.2.3.1.1 NexTrak <sup>TM</sup> Attendance Management System.....	14
2.2.3.1.2 Viking Attendance Management System.....	15
2.2.3.1.3 Jupiter for Window Time and Attendance System.....	17
2.2.3.2 Study on Current Software Development Tools.....	20
2.2.4 Group Discussion / Brainstorming.....	21
2.2.5 Book References.....	21



2.3	Conclusion.....	21
<b>CHAPTER 3 : SYSTEM ANALYSIS AND REQUIREMENTS.....</b>		<b>23</b>
3.1	Project Description.....	23
3.2	System Enhancement.....	23
3.3	New Functionalities and Features.....	24
3.4	Development Model.....	25
3.4.1	Approach.....	25
3.4.2	Justification of Spiral Model.....	26
3.5	Development Strategy.....	27
3.5.1	Functional Requirements.....	27
3.5.1.1	Functional Requirements for User Module.....	28
3.5.1.2	Functional Requirements for Administrator Module.....	31
3.5.2	Non-Functional Requirements.....	32
3.6	Development Tools.....	34
3.6.1	Reasons of Remaining ASP Technology.....	35
3.6.2	Reasons of Remaining SQL Database Server.....	35
3.7	Conclusion.....	35
<b>CHAPTER 4 : SYSTEM DESIGN.....</b>		<b>36</b>
4.1	System Functional Design.....	36
4.1.1	System Architecture.....	36
4.1.2	Data Flow Diagram.....	38
4.1.3	Data Process Flow for AMS and LMS Integration.....	40
4.1.4	System Structure Chart.....	41
4.2	Network Design.....	43
4.3	Database Design.....	44
4.3.1	Database Structure.....	44
4.3.2	Data Dictionary.....	45
4.3.3	Job Scheduling.....	49
4.4	Graphical User Interface Design.....	51



4.4.1	Design of Screen.....	51
4.4.2	Design of Output.....	52
4.5	Conclusion.....	52
<b>CHAPTER 5 : SYSTEM IMPLEMENTATION.....</b>		<b>53</b>
5.1	Platform Development.....	53
5.1.1	Window NT Server 4.0 .....	53
5.1.2	Internet Information Server.....	54
5.1.3	Microsoft SQL Server 7.0 .....	54
5.1.4	Microsoft Exchange Server 5.5 .....	62
5.2	Module Development.....	63
5.2.1	User Module.....	63
5.2.1.1	Stand-alone System.....	64
5.2.1.2	Web-based System.....	70
5.2.2	Administrator Module.....	73
5.2.2.1	Web-based System.....	73
5.2.3	Integrated Module.....	76
5.3	Conclusion.....	77
<b>CHAPTER 6 : TESTING.....</b>		<b>78</b>
6.1	Unit Testing.....	78
6.1.1	Code Reviewing.....	78
6.1.2	Input Testing.....	80
6.1.3	Data Flow Testing.....	82
6.2	Module Testing.....	84
6.3	Interface Testing.....	86
6.4	Integration Testing.....	86
6.5	System Testing.....	90
6.6	Conclusion.....	93



<b>CHAPTER 7 : PROJECT FINDINGS AND CONCLUSION.....</b>	<b>94</b>
7.1 System Evaluation.....	94
7.1.1 Review on Goal.....	94
7.1.1.1 Proof of AMS Feasibility.....	94
7.1.1.2 Achievement of Objectives.....	94
7.1.1.3 Foundation for Future Growth & Enhancement.....	94
7.1.2 System Strength.....	95
7.1.3 System Limitation.....	97
7.2 Problems Encountered.....	97
7.2.1 Setting Up the Operating System.....	97
7.2.2 Setting Up SQL Mail Services.....	98
7.2.3 Creating ActiveX dll File to Convert Datetime Format in ASP.....	98
7.2.4 Transferring the Existing Database to the New Database.....	98
7.2.5 Accessing the Session Value from Client Side.....	99
7.3 Recommendations and Future Enhancements.....	100
7.4 Knowledge Gained.....	100
7.5 Conclusion.....	101
<b>REFERENCES.....</b>	<b>102</b>
Figure 5-8 Completion of Job Schedule.....	60
Figure 5-9 Job Notifications.....	61
Figure 6-1 Invalid Login ID.....	60
Figure 6-2 Invalid Password.....	61
Figure 6-3 Invalid Date Selection.....	62
Figure 6-4 Enter Reason for Late.....	62
Figure 6-5 Check In Time Confirmation.....	63
Figure 6-6 Conversion of 200 Minutes Overtime.....	64
Figure 6-7 Updated Conversion of Overtime to Money.....	65
Figure 6-8 Unapproved Overtime to Money Conversion.....	65
Figure 6-9 AME Administrator's Welcome Page.....	67
Figure 6-10 User's Welcome Page.....	67



**LIST OF FIGURES**

<b>Figure 1-1</b>	<b>Project Schedule.....</b>	<b>8</b>
<b>Figure 3-1</b>	<b>Attendance Management System Development Model.....</b>	<b>26</b>
<b>Figure 4-1</b>	<b>AMS Three-Tier Client / Server Architecture.....</b>	<b>36</b>
<b>Figure 4-2</b>	<b>AMS Data Flow Diagram.....</b>	<b>38</b>
<b>Figure 4-3</b>	<b>AMS &amp; LMS Integration Data Process Flow.....</b>	<b>40</b>
<b>Figure 4-4</b>	<b>AMS System Structure Chart.....</b>	<b>41</b>
<b>Figure 4-5</b>	<b>AMS Network Setup Design.....</b>	<b>43</b>
<b>Figure 4-6</b>	<b>Database Structure.....</b>	<b>44</b>
<b>Figure 4-7</b>	<b>AMS Interface Design.....</b>	<b>51</b>
<b>Figure 4-8</b>	<b>AMS Output Design.....</b>	<b>52</b>
<b>Figure 5-1</b>	<b>SQL Server Agent Root.....</b>	<b>55</b>
<b>Figure 5-2</b>	<b>Enter Name for New Job.....</b>	<b>56</b>
<b>Figure 5-3</b>	<b>Before Step of New Job is Created.....</b>	<b>57</b>
<b>Figure 5-4</b>	<b>Putting Code into Command.....</b>	<b>58</b>
<b>Figure 5-5</b>	<b>After Step of New Job is Created.....</b>	<b>58</b>
<b>Figure 5-6</b>	<b>Schedule the New Job.....</b>	<b>59</b>
<b>Figure 5-7</b>	<b>Recurring Job Schedule.....</b>	<b>60</b>
<b>Figure 5-8</b>	<b>Completion of Job Schedule.....</b>	<b>60</b>
<b>Figure 5-9</b>	<b>Job Notifications.....</b>	<b>61</b>
<b>Figure 6-1</b>	<b>Invalid Login ID.....</b>	<b>80</b>
<b>Figure 6-2</b>	<b>Invalid Password.....</b>	<b>81</b>
<b>Figure 6-3</b>	<b>Invalid Date Selection.....</b>	<b>82</b>
<b>Figure 6-4</b>	<b>Enter Reason for Late.....</b>	<b>82</b>
<b>Figure 6-5</b>	<b>Clock In Time Confirmation.....</b>	<b>83</b>
<b>Figure 6-6</b>	<b>Conversion of 200 Minutes Overtime.....</b>	<b>84</b>
<b>Figure 6-7</b>	<b>Updated Conversion of Overtime to Money.....</b>	<b>85</b>
<b>Figure 6-8</b>	<b>Unapproved Overtime to Money Conversion.....</b>	<b>85</b>
<b>Figure 6-9</b>	<b>AMS Administrator's Welcome Page.....</b>	<b>87</b>
<b>Figure 6-10</b>	<b>User's Welcome Page.....</b>	<b>87</b>



<b>Figure 6-11</b>	<b>Approver's Directed Page.....</b>	<b>88</b>
<b>Figure 6-12</b>	<b>LMS Administrator's Directed Page.....</b>	<b>89</b>
<b>Figure 6-13</b>	<b>Overtime to Money Conversion Application List.....</b>	<b>90</b>
<b>Figure 6-14</b>	<b>Generate Attendance Report.....</b>	<b>91</b>
<b>Figure 6-15</b>	<b>Attendance Report.....</b>	<b>92</b>
<b>Figure 6-16</b>	<b>Attendance Summary Report.....</b>	<b>93</b>
Table 4-3	Activity Table.....	46
Table 4-4	Half/Leave Table.....	47
Table 4-5	Currency Table.....	48
Table 4-6	AttendanceOT Table.....	48
Table 4-7	Tar/TimeSheet Table.....	49
Table 6-1	Non-working Date.....	81
Table 6-2	Attendance Table.....	83
Table 6-3	Before Overtime to Money Conversion.....	84
Table 6-4	After Overtime to Money Conversion.....	90
Table 6-5	Matching Data in Database.....	92



**LIST OF TABLES**

<b>Table 4-1</b>	<b>Attendance Table.....</b>	<b>45</b>
<b>Table 4-2</b>	<b>Yearly Attendance Table.....</b>	<b>45</b>
<b>Table 4-3</b>	<b>HotLink Table.....</b>	<b>46</b>
<b>Table 4-4</b>	<b>Calendar Table.....</b>	<b>46</b>
<b>Table 4-5</b>	<b>Activity Table.....</b>	<b>46</b>
<b>Table 4-6</b>	<b>HalfLeave Table.....</b>	<b>47</b>
<b>Table 4-7</b>	<b>Cmoney Table.....</b>	<b>48</b>
<b>Table 4-8</b>	<b>AttendanceOT Table.....</b>	<b>48</b>
<b>Table 4-9</b>	<b>TstTimeSetUp Table.....</b>	<b>49</b>
<b>Table 6-1</b>	<b>Non-working Date.....</b>	<b>81</b>
<b>Table 6-2</b>	<b>Attendance Table.....</b>	<b>83</b>
<b>Table 6-3</b>	<b>Before Overtime to Money Conversion.....</b>	<b>84</b>
<b>Table 6-4</b>	<b>After Overtime to Money Conversion.....</b>	<b>90</b>
<b>Table 6-5</b>	<b>Matching Data in Database.....</b>	<b>92</b>



## CHAPTER 1 : INTRODUCTION

Ever since there have been organizations, there have been offices. Offices are the centers of management. They are needed to organize, process and handle the information that the organization needs in its day-to-day operations. Paper is used as a mean of communicating and storing massive amounts of data, information and knowledge. However, paper is inefficient because it is hard to update and the information on one piece of paper is hardly related to information on the other pieces of paper. Beside that, the clerk has to manage large volumes of documents, which often duplicate everyday.

Today's electronic technology continues to change the way people accomplish their work. Both the public and private sectors are gradually moving toward 'paperless' environment, which requires a minimal use of paper documents and reduces need for human handling of routine tasks. The purpose is to eliminate the inefficiencies that a paper-based, relatively labor-intensive environment causes, such as errors from duplicative data entry, and to automate routine functions if the benefits of doing so outweigh the costs. Thus, most but not all information would be transmitted and stored electronically.

Beside that, users have greater access to one another now, particular through the international system of computer networks known as Internet. This is due in part to greater interoperability such as the ability of diverse electronic systems to communicate with each other through any network. To fulfill the needs of the communication industries as well as the organizations, the idea of **Generic Office Environment (GOE)** has been come out by our Prime Minister. It is geared to take advantage of technology to support our daily needs to access information more efficiently and effectively.

GOE is said to provide a fully integrated, distributed, and scaleable paperless office environment by deploying multimedia information technology. Civil servants will be provided with a quick-and-easy method to access up-to-date, accurate information, and ensure that the right information gets to the right people at the right time.



Therefore, modern office environment requires the establishment of a technology infrastructure that provides connectivity and the necessary tools to get the job done easily. For examples, electronic swipe cards system and keypad controlled graphical time and attendance system are used in place of the traditional punch cards system. With this technology, employees' in and out time are recorded by computer, which can be located in another office hundreds of feet away. Employees' data are also easily stored with the advance database technology available in market such as Oracle database and SQL server. Consequentially, data requirement having evolved into information requirements, have been further delineated into corporate knowledge requirements.

### 1.1 Overview of Attendance Management System

As computer based information system is essential for most of organization to compete and gain the competitive advantage, the web based Attendance Management System (AMS) has been implemented in Faculty Computer Science and Information Technology (FCSIT) since 1<sup>st</sup> January of 2000 as an alternative to the punch cards system. This system is said to bring a lot of benefits to faculty in reducing the paper works, recourses and saving valuable time and cost.

AMS provides complete solutions of human capital especially in managing attendance policies of FCSIT towards the goal of Generic Office Environment. A secure, faster and easier way to clock in and clock out function is provided for user. Beside that, functions are customized for user to apply over time. The overtime can be converted on line to either money or half-day leave. Users also can generate different type of reports through this system.

AMS is running in a client / server environment with SQL server database and also as a stand-alone application. Enhancement of the system and implementation of new functionalities and features will be the focus throughout this project. This project will be integrated and carried out together with the Leave Management System (LMS). Therefore, FCSIT will be the test bed for these two systems.



## 1.2 Project Objectives

The existing Attendance Management System has limitations that need to be improved and enhanced. This project will focus on the adaptations of the system in the real and changing environment. Below are the project objectives:

- **To analyze and study the problems faced by users and administrators in the current system.**

Since the system has been implemented in faculty, users and administrators found that the system sometimes does not deliver the right information at the right time. The system behaves in a way that users did not expect. They faced many problems in using the system. The erroneous output that generated by the program may be caused by many factors. So, this project is carried out in order to find out the problems faced by them.

- **To find out the solutions for the facing problems and enhance them by adding more features or functionalities.**

There must be solutions for all the problems that have been identified. It is no point to identify the problems without any further actions. This project will try to solve the users' problems as well as introduce new functionalities and features that will bring benefits to them.

- **To enforce faculty policies mainly in attendance and leave policies in order to improve attendance and increase productivity dramatically.**

It is the vision of faculty that the AMS will assist in enforcing the attendance and leave policies. This project is hoped to ease the entire workflow of attendance process and keep track on those who are always late for work. For an example, a warning letter will be send to the late comer who has exceeded five times within a month. Beside that, the overtime rule could be set too to control how the overtime is calculated. Consequentially, it reduces the cost of faculty's absenteeism while at the same time result in productivity gains.



- **To maintain a central repository for all human resource and pay roll related information.**

Since there is a database for storing all the data and information of the staffs' particular other than attendance and leave records, it is hoped that the database can act and maintain as a central repository for all human resource and pay roll related information.

- **To make the system more user friendly and user defined without having the staffs to be the 'computer experts'.**

This project also sets the objective to make the current system more user friendly and user defined. The interfaces of the system are easily comprehended and users do not need to remember all the commands in order to use the system. What they need to do are just a few clicks on mouse to get the desired output. They can also customize the reports in the way they want to see them.

- **To enhance the integration with Leave Management System.**

The last objective of this project is to enhance the integration with Leave Management System (LMS). User just needs to enter password once to log in to both AMS and LMS. Currently, the same user is holding two different passwords in order to access to each system. So, the integration enhancement will be the focus under this project to achieve the goal of effectiveness and efficiency.



### 1.3 Project Scope

The implementation of attendance management system will require a significant of man-hours, so the project has to be defined so that the system can be completed in a limitation of time. For this project, the scope is only limited to enhance the system for the FCSIT.

It covers all the modules that fulfill the needs of faculty and users. There are two main modules involved, which are user and administrator modules.

#### 1.3.1 User Module

This module is designed for user to perform the basic attendance functions such as clock in, clock out, apply for overtime, generate personal attendance reports, view the daily attendance sheets, applying the backward overtime and convert the overtime to half-day leave or money. It is designed to meet the personal use too. Calendar sub systems will be provided to user for organizing their own activities and important events. There would be a list of address for hot sites grouped in different category for user to access.

#### 1.3.2 Administrator Module

While the administrator module is mainly designed for the purpose of administering and monitoring the attendance process of faculty. Different kinds of configurations can be done here such as attendance setting and working hours setting. Reporting is the important features under this module where it will be used for analysis purposes. Lastly the hot site browsing and adding is the new function provided for the advantage and convenient of administrators.

This system will include the academic staffs and also the non-academic staffs in FCSIT. Students are not allowed to use this system, however, they can only access to the home page of this system to view the daily attendance sheets. An authentication check is required before the user can access to AMS and LMS.



## 1.4 Project Motivation

By doing this project, not only the users are gaining a lot of benefits, it will also result in productive gain of faculty. Below are the reasons why this project is motivated:

### ➤ Current Modules are expandable

The modules under this system not only can be modified but also can be expanded. The coding is well structured and defined where the new functions can be easily written to the modules. They are not complicated and fixed like other systems. Thus, the enhancement of the system can be carried out smoothly.

### ➤ Speed out data processing, time and cost saving

It is said to speed up data processing by enhancing this system. The overall revision and review process can be shortened. As a result, it brings to time saving whereby users will have more time to do other tasks that are more productive. Other than that, faculty will save a lot of cost when there is an environment that can be produce more in less time at lower cost and with high quality. The cost of doing this project is reasonable and affordable by faculty. So, this project is feasible and should be moved on.

### ➤ Current system is scalable and inter-operable

The current AMS has the capability to migrate as a client or server to machines of greater or lesser power, depending upon requirements with little or no change to underlying components. This is very importance because the enhancement may need the bandwidth to be increased or additional web server to be implemented. Beside that, the current system is able to work with different types of application to share data and process. It can be integrated with LMS or other GOE system. This factor will definitely become one of the project motivations.



➤ **Higher Quality: More functionalities and services**

Users will be satisfied by the higher quality of the system. They are enjoying not only the basic functions of attendance but also more services from the system such as calendar & diary system.

➤ **Able to adapt in changing environment**

This is the last factor of the project motivations. It is useless to develop or enhance a system that cannot adapt in the changing environment. AMS has described the fundamental assumptions on which the system is based and anticipated changes due to hardware evolution, changing user needs, and so on. With the ability to adapt in the changing environment and new technologies, the system is motivated and worth to be enhanced.

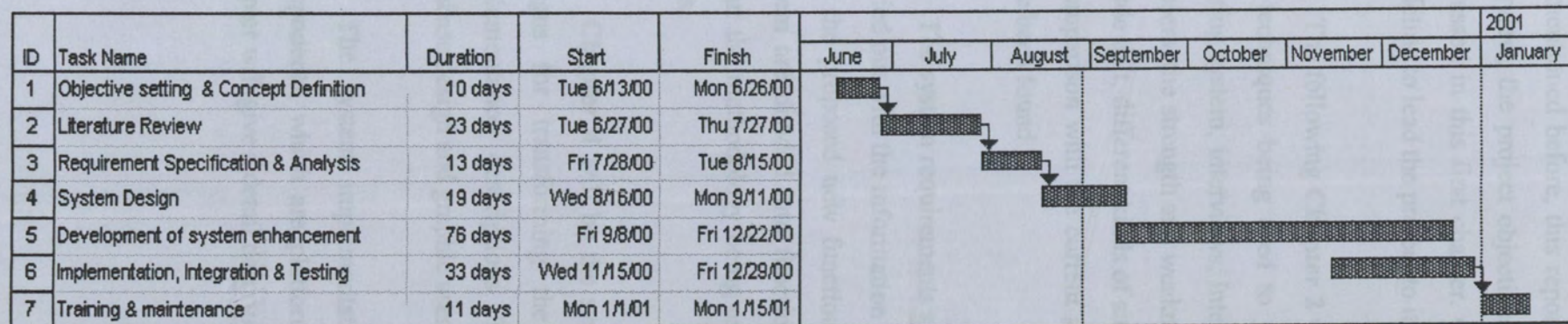
**1.5 Project Schedule**

This project is scheduled based on the activities so that the works can be completed on time. It is started on 13<sup>th</sup> June 2000 until 15<sup>th</sup> January 2000. The schedule of the project is shown in Figure 1-1.

Figure 1-1 Project Schedule

Task Name	Duration	Start	Finish
1. Requirements Gathering & Concept Development	10 days	Fri 01/06/00	Mon 05/06/00
2. System Analysis	20 days	Tue 06/06/00	Thu 26/06/00
3. Software Requirements & Analysis	15 days	Fri 09/06/00	Tue 20/06/00
4. System Design	10 days	Wed 21/06/00	Mon 04/07/00
5. Development of system software	30 days	Fri 09/07/00	Fri 12/08/00
6. Implementation, Testing & Training	30 days	Wed 11/08/00	Fri 12/09/00
7. Deployment & Evaluation	10 days	Mon 15/09/00	Mon 15/10/00





**Figure 1-1 Project Schedule**



## 1.6 Report Organization

As mentioned before, this report is focusing on the enhancement of AMS. The project overview, the project objectives, the project scopes and project motivation have been discussed in this first chapter. Other than that, the project schedule is drawn out as a guideline to lead the project to its targeted dateline.

The following **Chapter 2** will be the literature review. This chapter will talk about the techniques being used to gather idea and information. Examples are review of existing system, interviews, Internet research and brainstorming. All the problems faced by users, the strength and weaknesses of the current system are identified in this chapter. Beside that, different kinds of similar systems are referred and reviewed in order to make a comparison with the current system. The end of this chapter will summarize of what have been found.

The system requirements and analysis will be elicited in **Chapter 3**. The analysis is carried out with the information gathered from the review of literature. The enhancement and the proposed new functionalities will be covered here. The requirements of the system are divided into functional and non-functional. This chapter will also discuss about the methodology being used, justification of the chosen development model and tools.

**Chapter 4** will be the system design. This chapter will cover all the associated designs for transforming the requirements identified in chapter 3 to detailed implementation descriptions. They are system functional design, network design, database design and graphic user interface design.

The system implementation is explained in **Chapter 5**. It involves two components, which are platform development and the modules implementation. This chapter will give a detail description on each component of system implementation.



Project testing will be covered in **Chapter 6**. There are five types of testing that have been carried out under this system. They are unit testing, module testing, interface testing, integration testing and lastly is the system testing.

**Chapter 7** is taking about the project findings and conclusion. Under this chapter, the system is evaluated and all the problems encountered are discussed here. Beside that, recommendations to the system as well as future enhancements, knowledge gained from the project and review on goal is covered in this chapter. Lastly, there would be a conclusion for the whole project.

Following are the techniques that have been used in the review of literature:

- Review of the existing system
- Interview
- Internet Research
- Group Discussion / Brain Storming
- Book References

### 2.2.1 Review of the Existing System

The current Attendance Management System is divided into two applications, which are the web-based application and the stand-alone application. The stand-alone application is written in Visual Basic 6.0 and is designed for the use of non-academic staff. It is installed in the computer located in the front door of old building in PCSTT. The academic staffs are not allowed to use the stand-alone machine. They are only allowed to access to web-based application through their personal computer. However, non-academic staffs can access to both of these applications.

The web-based application can be accessed through <http://ams>. The two main modules involved in this system are user and administrator modules. Each of the modules performs different type of functions. The system functional requirements and non-functional requirements have been examined in order to understand the usage of the system.



## CHAPTER 2: LITERATURE REVIEW

### 2.1 Purpose

Before start the project, a literature review should be carried out. The purpose is to gain an understanding of current Attendance Management System, study and analyze its problems as well as to gather information and idea in developing and enhancing the system. Without this understanding, effectiveness system design is impossible.

### 2.2 Approach

Following are the techniques that have been used in the review of literature:

- Review of the existing system
- Interview
- Internet Research
- Group Discussion / Brain Storming
- Book References

#### 2.2.1 Review of the Existing System

The current Attendance Management System is divided into two applications, which are the web-based application and the stand-alone application. The stand-alone application is written in Visual Basic 6.0 and is designed for the use of non-academic staffs. It is installed in the computer located in the front door of old building in FCSIT. The academic staffs are not allowed to use the stand-alone machine. They are only allowed to access to web-based application through their personal computer. However, non-academic staffs can access to both of these applications.

The web-based application can be accessed through <http://lms>. The two main modules involved in this system are user and administrator modules. Each of the modules performs different type of functions. The system functional requirements and non-functional requirements have been examined in order to understand the nature of the system.



Through out the review, it is found that the system is still lacking some of the functionalities and inefficient in its daily office processing. The system has been tested and run with some of the sample data. For further analysis, its coding is revised. All the problems faced by the system are identified and examined from a different angles and viewpoints as these provide different insights into the system requirements.

### 2.2.1.1 The System Deficiency and Weaknesses

Below are the deficiency and weaknesses of the system that have been identified:

- Daily updates of attendance records are performed by stand-alone application. It will create the 83 dummy records in database every morning. When the staffs are coming to work and clock in their attendance, their record will be searched and then be updated. This process is said to slow down the data processing of the system and also unsecured since any one can just plug off the cable of the machine or switch off the main power of the machine and reboot it.
- Each query in the current system is performed by independent Transact SQL Command and is send to database through the Internet Information Server (IIS). It is said to increase the network traffic when hundreds of Transact SQL are sent over the network at the same time.
- The size of the database is also keep increasing day by day. There aren't any archive functions provided in current system to manage space and tune the performance of the database.
- The daily attendance records are viewable to staffs of FCSIT but not to the students. Currently, students look for lecturers in their respective rooms without ensuring the lecturers are in or not. If the lecturers were on leave, it would be a waste of time for students to check their availability all the time.
- Activities application is not utilized using reminder or calendar and therefore the availability of a lecturer is hard to trace.
- Non-academic staffs are allowed to convert their overtime to half-day leave. There is only one type of half-day leave, which is 'whole day'.



- Non-academic staffs are allowed to take overtime without limitation or approval from their head department. This will result in the loss of faculty if they are doing other things, which are not relevant to their jobs.
- The current system is not performing a function to keep track, detect or controlling those who are late for work. There aren't any warnings letters send to latecomers. Indeed, there will be a letter to remind them to clock in if they forget to do so after 12.00 pm.
- The reporting system is very confusing and is not designed according to its types.
- Same user has to log on to the system twice to access to both AMS and LMS. It is inconvenient and inefficient.

### 2.2.2 Interview

Interviews are also the main resource in understanding the current system especially in getting feedbacks from users. A few interviews have been carried out. Firstly, an interview had been carried out with the administrator of this system, Puan Azlin. Then, it was followed by the interview with the Deputy Dean, Dr. Diljit Singh and one of the head departments, Dr. Selva. They had stated out the system problems and its limitation.

Lastly, the office clerks were interviewed to get the feedbacks about the current system. The supervisor of this project, Mr. Ling Teck Chaw had also given some valuable advice and updated information that is very important in enhancing the system.

Below are the feedbacks from users and administrator:

- To make the over time application more flexible and not only fixed to after working hours (5.00 pm – 11.00pm)
- All the overtime applications need to be approved by the first approver or second approver from their own department.
- To add a function of overtime conversion, allowing the non-academic staffs to convert their overtime to half-day leave or money.
- To have an ease use and friendly interface.
- To improve the efficiency and reliability of the system.



### 2.2.3 Internet Research

Internet research is carried out for two purposes – a) study on similar attendance management systems and b) study on current software development tools. Many of the related sites have been searched and referred in order to collect information and idea.

#### 2.2.3.1 Study on Similar Attendance Management Systems

There are many attendance management systems available in the Internet such as NexTrak™ Attendance Management System, Viking Attendance Management System and Jupiter for Windows Time and Attendance System.

All the features and functionalities of these systems will be compared with the current Attendance Management System. However, all of these features are only taken into consideration and their feasibilities have to be tested before being implemented. Below are some of the samples of the similar attendance management system found in Internet.

##### 2.2.3.1.1 NexTrak™ Attendance Management System

<http://202.233.127.171/nextrakattendancemanagementsy.htm>

<http://202.233.127.171/Product2.htm>

NexTrak™ Attendance Management software from Kronos automates virtually any attendance program, including policies using points, occurrences, hours missed, or percentage of absenteeism. An unlimited number of policies, the rules each policy contains, what actions that the software needs to perform, and the documentation that want to be generated can be defined. NexTrak collects the required data from the Timekeeper system, applies the user-customized rules to the absences and exceptions, and ensures they are uniformly applied.



### **Features of NexTrak™ Attendance Management System**

#### **➤ Application Flexibility**

It allows the system to be used in a client / server environment with an Oracle or SQL Server database, or as a stand-alone application.

#### **➤ Attendance Policy Customization**

It permits a variety of attendance and disciplinary rules to be combined into an unlimited number of administered policies. Calculation methods, accumulators, actions required and levels of severity are all specified to meet the requirements of organization's policy definition. The system will give an alert when action is required and generate the appropriate disciplinary notices to help ensure they are issued in a timely fashion. These notices are easily customized using the document editor, and may even be archived for later retrieval and reprinting.

#### **➤ User-definable Data**

It allows fields to be added to the employee database for use as additional policy, grouping or report criteria.

#### **➤ Dynamic Reporting**

It provides powerful querying and report generation tools, giving users the data they need, the way they want to see it. The built-in *Graph Manager* provides standard, cross-tabulated and statistical queries for comprehensive data analysis presented in a variety of 2D and 3D graph formats. Data query results may also be exported to the most popular formats, including spreadsheet, database and text files.

### **2.2.3.1.2 Viking Attendance Management System (VAMS)**

<http://www.vikingsolutions.com/>

VAMS helps organizations **manage** the attendance of employees. By focusing on problem areas and/or employees, and through various "alerts", VAMS allows organizations to take **pro-active** measures to improve attendance and increase productivity dramatically.



Absenteeism, in all its facets, short and long term disability, occasional sick days, Workers' compensation, vacation, training, days off in lieu of overtime, special and family leave (as per FMLA, the Family Medical Leave Act), must be accurately recorded, tracked and costed, in order to be in a position to MANAGE the process. VAMS accomplishes the first phase of recording, tracking and costing the attendance of employees easily and accurately (more accurately than most systems), without having staff to become "computer experts". It is extremely user-friendly and user-defined.

### **Features and benefits of VAMS**

- Extensive client customization (absence codes, position, employee status, classification, security access, etc.)
- Multi-schedules with easy set-up and system prompts
- Cost factors to calculate direct and indirect costs
- Custom organization structure - dynamic up to 12 levels
- Automatically generates letters for attendance management
- Seamless integration with word processing and spreadsheet packages
- Import/export capabilities
- Ease of data entry - individual or multiple employees
- Effective attendance management reporting including graphs
- Maintains employee absence data in organization at various locations, before and after transfers, promotions, etc.
- Identifies high users by absence type
- Superior reports to identify and encourage good performers
- System generated prompts to supervisors to ensure follow-ups
- Corporate overview - up-to-date corporate wide information with a simple entry
- Comparative data reports for any period
- Employee profile - complete historical data enabling comparisons between time periods and peer groups
- Personal information manager for each supervisor
- System generated alerts for high leave usage at client defined levels
- Full year at a glance for on screen analysis



- Archiving capabilities
- All reports can be viewed on screen before printing
- Local Area Networks
- Automatic leave banking
- Up to five different customized leave banks
- Instant on screen balances
- System generated alerts when banks per employee drop below pre-determined levels

### **2.2.3.1.3 Jupiter for Windows Time and Attendance System**

[http://www.jantek.com/frames/jta\\_startup.htm](http://www.jantek.com/frames/jta_startup.htm)

The Jupiter for Windows Time and Attendance System is an integrated software / hardware combination that allows companies total control over their employees' time and attendance data. The system offers unparalleled performance, with the flexibility and functionality required for today's business environment.

Jupiter for Windows automatically consolidates, calculates and processes information for reporting and communicating directly to outside payroll service or in-house payroll system. In addition it will generate all kinds of management reports upon which informed, timely decisions can be made.

#### **Features of Jupiter for Windows Time and Attendance System**

##### **➤ Employees**

Employee setting up is done easily by importing them from existing employee databases such as payroll service packages or human resource systems. Or key the information in manually at the conveniently formatted entry screen. The Employee Edit Screen groups together in pages all related information and provides links to other setup screens. The entry screens also allow input of multiple pay rates. With a digital camera or scanner, pictures with employee files for positive identification and badge printing purposes can be associated.



➤ **Policies**

It offers maximum flexibility in setting up pay policies, covering as many rules as possible. Basic rules are grouped together in screen pages to avoid the difficulty of sorting through strictly optional rules before the system can be up and running.

➤ **Attendance Management**

On occasion, timecards (for example when an employee forgets to punch IN or OUT) may need to be edited. The Time Card Edit screen gives the complete tools to quickly locate exceptions that require correction. The Time Card Edit screen displays all punches and calculated hours in a convenient, easy to understand format. Buttons at the bottom of the screen allow immediate access to editing windows and supporting information. The exception report button provides links to all employee records, simplifying correction procedures. When corrections are made to a timecard, all hours are recomputed, and required adjustments for daily and weekly overtime are applied automatically.

➤ **Scheduler**

Schedules can be set up for individual employees or created for entire departments or job designations. They can be based on employee's normal shifts, or copied from weekdays, other weeks or other employees. A quick reference chart shows the projected daily schedules to ensure proper staffing during peak and non-peak hours. Calendar-type reports give employees and supervisors quick, graphic looks at their schedules for them to note accordingly.

➤ **Auto Procedure**

Jupiter for Windows can be set to execute predetermined instructions such as data collection and reporting at certain times of the day, such as after office hours or when the facility is closed down. Procedures can be set to run hourly, daily, weekly, monthly, or for specific dates and times.



➤ **Accruals**

The Accruals module automatically calculates benefit hours such as sick and vacation time based on company's policies and restrictions. The system performs availability checks when entering these hours at the time card edit screen, preventing the use and payment of hours not yet earned. Accrual reports display the number of hours accrued, earned, used and available for review by supervisors and management.

➤ **Report Generator**

Jupiter for Windows' Report Generator offers a selection of pre-formatted reports. Report categories include Time and Attendance Reports, Labor Reports, Management Reports, Human Resource Reports, Exception Reports, Scheduling Reports and Audit Trails Reports. Within each of these header categories there is a further breakdown of report selections. Definable parameters allow the coverage of reports to be limited.

➤ **Archiving**

The archiving feature compresses and stores old time card information, freeing valuable hard drive space and reducing system-processing time by moving data off-line. All archived information can be retrieved, viewed and printed in any of the standard report formats.

➤ **Payroll Interfaces**

Jupiter for Windows has over 150 payroll interfaces available, allowing connectivity to company's payroll service provider or in-house payroll system. Interfaces assure that data transfers are 100% accurate and save valuable employee time. Interfaces can be set to transfer shift differential, labor distribution and job tracking information if required. Payroll details can be viewed and changed before final conversation and transmission.

➤ **Lockout**

The lockout module allows managers to restrict employee punches to certain times of the day. Default lockout times can be set for all employees for the entire facility, or



for individual employees following their shifts and schedules, controlling access and limiting unauthorized overtime.

➤ **Backup**

A complete copy of all data files can be backed up and stored in compressed format for convenience and portability. When restoring data to later software versions, the backup data structure is automatically adjusted to match the current version.

➤ **Security**

Jupiter for Windows provides screen level, window based security. Each screen may be blocked off for selected users, and details such as dollar values and personal information may be made invisible. Users' views may be restricted to specific groups of employees. The system also maintains full audit trails, allowing accountability and recreation of activities for administrative purposes.

➤ **On-line Help**

Jupiter for Windows comes with on-line, context sensitive help screens, including detailed explanations for each screen as well as detail setup procedures. Each help-screen also comes with the actual screen set to help users understand each data field.

### 2.2.3.2 Study on Current Software Development Tools

The technologies and software development tools used under this existing system is re-examined. The purpose is to find out whether they are suitable or need to be changed. Internet research has been carried out for all the related tools such as SQL Server, Active Server Page and Internet Information Server (IIS). Below are the links that have been referred:

<http://www.asp-zone.com/>

<http://www.asptoday.com/articles/19990601.htm>

<http://java.sun.com/docs/books/tutorial/jdbc/basics/sql.html>

[http://msdn.microsoft.com/library/psdk/sql/8\\_des\\_07\\_2.htm](http://msdn.microsoft.com/library/psdk/sql/8_des_07_2.htm)

<http://www.microsoft.com/iis/default.asp>



### 2.2.4 Group Discussion / Brainstorming

In order to come out as many ideas as possible and find out the solutions to the current system, there was a group discussion with the developer of LMS, the successor of this system and also the supervisor of this project. All the problems faced by current system were discussed and the requirement analysis was done together. Brain storming section was also carried out.

### 2.2.5 Book References

Throughout the process of understanding the current system, different kinds of books, magazine and newspapers have been referred. Examples of the reference are 'Active Server Page 3.0', 'SQL server 7.0', 'IT world' and 'Using Microsoft Exchange server'. All the references are useful in understanding and figuring out the coding of the system so as to elicit the system requirements.

## 2.3 Conclusion

After going through the literature review, it is found that the current system is still inefficient in its daily data processing. Enhancement should be carried out to solve the problems faced by user and administrator. The deficiency and weaknesses of the system, which are described in section 2.2.1.1 can be solved by adding new features or functionalities.

The feedbacks from users and administrator are useful to the enhancement of this system. Beside that, many new ideas are found through Internet research, especially from the studies on other organization's attendance management system in section 2.2.3.1. All the new features that the current system is lacking of are taken into consideration. However, there are still have constraints in implementing the new found features. This is because not all of the new features are suitable for the current system. Below are some of the features and reasons why they are rejected:



- The badge readers, optical scanners or swipe card system are too costly.
- Function of handling multiple shifts is not appropriate since FCSIT is an academic organization, which does not provide shift work.
- The payroll system of FCSIT is an independent entity under the control of human recourse department of University Malaya. It is hard to link to the AMS. The system will be too large and is doubtful to complete it in such a short time.

From the study, it is found that the technologies and developments tools used under this system are compatible and reliable. All these findings will be the basis for the new system enhancement. The enhancement of this system will be described in more details in the following chapter.

- Perform daily scheduled job in creating attendance records through SQL server, and through the stand alone application as done previously. The job will only be triggered if the particular day is a working day.
- Perform daily scheduled job in deducting leave and half-day leave in SQL server, which will be updated automatically in the daily on-line attendance sheet.
- Perform daily archive function in SQL database to control the growth of database. Separate the attendance table to daily and yearly and transfer all the data in daily to yearly after 30 days.
- Perform the yearly scheduled job in deleting all the good attendance records in Yearly Attendance table on the end of December.
- Use the store procedures in retrieving or manipulating data in database to replace the current independent Transact SQL statements.
- To extend the time of overtime from 11.00 pm to 12.00 pm and also allow the staff to apply over time before working hours (6.30 am-8.30 am), during the recess time on Friday (12.45 pm - 2.45 pm) or any other available times.
- All the overtime applications will be approved by first or second approver according to their own department. This module is integrated in the leave management system.



## CHAPTER 3 : SYSTEM ANALYSIS AND REQUIREMENTS

### 3.1 Project Description

This project will be covered in two parts. The first part is the **enhancement of the system**, while the second part is the **new functionalities and features** adding.

### 3.2 System Enhancement

System enhancement will involve all the modules of the current system. It would be carried out based on the system deficiency and weaknesses that were identified in **section 2.2.1.1** and the feedback from users and administrator as described in **section 2.2.2**. Some of the modules' components need to be enhanced or dropped out from the system. Below are the enhancements to the system that would be carried out:

- Perform daily scheduled job in creating attendance records through SQL server, not through the stand alone application as done previously. The job will only be triggered if the particular day is a working day.
- Perform daily scheduled job in detecting leave and half-day leave in SQL server, which will be updated automatically in the daily on-line attendance sheet.
- Perform daily archive function in SQL database to control the growth of database. Separate the attendance table to daily and yearly and transfer all the data in daily to yearly after one day.
- Perform the yearly scheduled job in deleting all the good attendance records in Yearly Attendance table on the end of December.
- Use the store procedures in retrieving or manipulating data in database to replace the current independent Transact SQL statements.
- To extend the time of overtime from 11.00 pm to 12.00 pm and also allow the staff to apply over time before working hours (6.30 am-8.30 am), during the recess time on Friday (12.45 pm – 2.45 pm) or any other available times.
- All the overtime applications will be approved by first or second approver according to their own department. This module is integrated in the leave management system.



- Add two more types of half-day leave, which are 'Morning Half-Day Leave' and 'Afternoon Half-Day Leave', other than 'Whole Day'.
- Reports will be generated in a more systematic way, which is according to specific type of reports, rather than group all the reports in one screen. Users can view the summary of their report too.
- To change and improve the interface of the system since some of the buttons are confusing.
- To detect third Saturday as holiday through the system.
- To change the working hour set up since the half-day leave function is added to the system.
- To have a fully integrated interface with Leave Management System. Users need to log on into the system once only for the both system.

### 3.3 New Functionalities and Features

The AMS is refined in ordered to meet the requirements of users as well as faculty. From the review done in **section 2.2.3.1.1**, **section 2.2.3.1.2**, and **section 2.2.3.1.3**, it is found that there are many features and functionalities of other similar attendance management systems that can be applied in this project. They have been recognized after comparing with the current AMS. Some of the idea is come from group discussion and brainstorming. Below are the new features and functionalities that would be added to the system:

- Calendar, which acts as a diary for users to put in details of their activities.
- Activity reminder that can be customized by users and will be popped up as users log into the system if the reminder is set for that particular day.
- Auto-generated warning letter to those who are late for work.
- Backward overtime applications, which allows non-academic staffs to apply the previous overtime that they have forgotten to log in.
- Overtime conversion, which allow non-academic staffs to convert their overtime either to half-day leave or money.
- Hot site browsing and adding.



### 3.4 Development Model

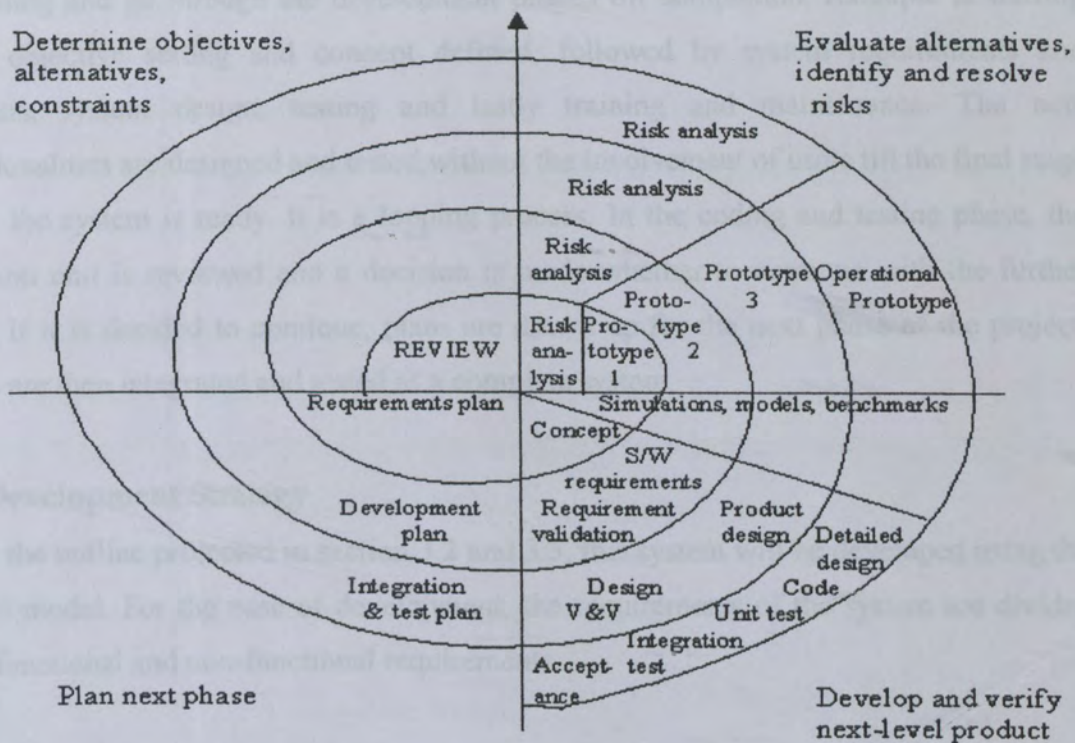
Process model is very important during the development of software. It can form a common understanding of the activities, resources and constraints involved in software development. When a process model is created, it helps in finding the inconsistencies, redundancies and omissions in the process.

#### 3.4.1 Approach

The approach that will be used throughout this project is the **Boehm's Spiral Model**. This approach takes the form a spiral where each loop represent phase in software development process. There is no need to adopt a single model in each cycle of the spiral or, indeed, for the whole of one software system. The spiral model encompasses other process models. Prototyping may be used in one spiral to resolve requirements risk. This may be followed by a conventional waterfall development. Formal transformation may be used to develop those of the system with high security requirements and a reuse-oriented approach used for implementing the other parts of the system.

Figure 3-1 shows the Attendance Management System development model. As mentioned above, this model encompasses other process models. Thus, it is a mixture of Waterfall, Prototyping and Incremental development models. Waterfall model concept is applied during the earlier stage. Then, the prototyping and incremental concept is added during the modules development.





**Figure 3-1 Attendance Management System Development Model**

### 3.4.2 Justification of Spiral Model

The Spiral Model is used because it meets the objectives of the project in enhancing the system as well as developing the new functionalities and features. In the aspect of enhancement, the concept, purpose and functionality of the current AMS are identified and defined. Then, system requirements are reviewed so that the enhancement can be carried out. There is an iterative and frequent feed back from users in the system design stage. The incremental prototyping method will be used based on each module. The system is modified and keeps changing till the users' specific requirements are met. It is a looping process and serves as a basic to derive a system specification. This method may address the user needs and expectations more closely.



While in the aspect of adding new functionalities and features, it will start from the beginning and go through the development phases till completion. Example is starting from objective setting and concept defined, followed by system requirements and analysis, system design, testing and lastly training and maintenance. The new functionalities are designed and tested without the involvement of users till the final stage when the system is ready. It is a looping process. In the coding and testing phase, the function unit is reviewed and a decision is made whether to continue with the further loop. If it is decided to continue, plans are drawn up for the next phase of the project. Units are then integrated and tested as a complete system.

### 3.5 Development Strategy

With the outline projected in section 3.2 and 3.3, this system will be developed using the Spiral model. For the ease of development, the requirements of the system are divided into functional and non-functional requirements.

#### 3.5.1 Functional Requirements

The functional requirements take into account the user and administrator modules. In this project, the functional requirements being concerned are in the aspect of enhancement and new functionalities, which have been identified.

Since the AMS and LMS have been integrated into one system, users, approvers and administrators from the both system will access into the system through the same page. They will then be directed to different page based on their access right. Below are the functional requirements pre-access into the system:

##### ➤ Log In

All users, approvers and administrators need to provide login ID and password for verification in order to access into the system. All users of AMS or LMS and administrator of AMS will be directed to welcome page. They will be provided with an option menu on the left and different category of internet browsing addresses on



➤ the right within this welcome page. While for the approvers and administrator of LMS, they will be directed straight to the leave management system.

➤ **Change Password**

Users can change their password before log on into the system as long as they are the authorized users.

➤ **View Daily Attendance**

➤ This link will display the daily attendance record for 3 types of users, who are academic staffs, non-academic staffs and tutors. Beside that, it will display the list of overtime and activity for that particular day.

### 3.5.1.1 Functional Requirements for User Module

➤ **Clock In**

This link will appear in the welcome page as short cut link and in the attendance page. It allows user to clock in for daily attendance. If the user is late, he or she needs to provide the reason before the attendance record is saved into the database. After that, the clock in link will be replaced by the clock out link.

➤ **Clock Out**

This link let the user to clock out after working hour, if the user clock out before working hour, the user will be prompted to enter the reason and his or hers logout status will become illegal. If the user clock out after working hour, the status will be good.

➤ **Attendance**

User gets into the attendance page by clicking on this link.



➤ **Leave**

User gets into the leave page by clicking on this link.

➤ **Calendar**

User can put in their current activities or events as well as their planned event in the calendar. It acts as a diary. From this calendar, an add reminder link will be appeared in the bottom of the calendar if the chosen date is today or coming date.

➤ **Reminder**

It is used to remind the users about their activities by popping out a message when the user log into the system. Users need to key in their event title, event location, date, time and so on. Users can customize their activities reminder, either to the known of personal, public or both. There is a link provided to delete the coming reminder too.

➤ **Overtime Clock In**

User, who wants to work overtime, needs to overtime clock in by clicking on the link. The user need to specify the reason for overtime and is allowed to overtime clock in only on the particular times, which are:

1. Monday to Friday

Before working hours (6:00 am - 8:00 am)

After working hour (5.00 pm- 12:00 am)

2. Friday

During the recess time (12:00 pm - 2:45 pm)

3. Saturday

Before working hours (6:00 am - 8:00 am)

After working hour (1.00 pm- 12:00 am)

➤ **Overtime Clock Out**

User need to clock out for overtime, otherwise their application would not be send to approver for approval



### ➤ **Backward Overtime**

User can apply the previous overtime that they have forgotten to log in. The application will be send to approver for approval. If the application is approved, the user will be notified through email. Anyway, user is allowed to delete any of the unprocessed overtime application through the link provided at the bottom of the page.

### ➤ **Overtime Conversion**

Overtime conversion can be done in two ways, which are:

#### 1. Convert to half-day leave

The half-day leave conversion could only be allowed if the available overtime is sufficient and it is the coming half-day leave. User has the option to delete the unprocessed half-day leave conversion too.

#### 2. Convert to money

Overtime can be converted into money too only if the user has the sufficient overtime. The user can delete any of the unprocessed overtime to money conversions through the link provided in the bottom of the page.

### ➤ **Reports**

There are four type of reports available for user, which are attendance, leave, overtime and activity reports. User is allowed to view their personal reports not the others. There are options provided for the user either to view the report of today, in a range of date or monthly. Beside that, user can generate the reports based on other selections provided in the screen. He or she can view the summary reports too.

### ➤ **Hot site browsing**

This feature allows users to select the Uniform Resource Locator (URL) from the combo box to browse their desired site. The URL of the sites is group into groups such as email hot site and education hot site. Users have no right to add new URL to the combo box.



➤ **Daily Attendance**

The function of this link has been explained in the pre-access system.

➤ **Auto-generated Warning Letter – Late for Work**

Staffs are allowed to clock in for work on the plus and minus 5 minutes time basis. If they are late for work, they need to state out their reasons. Those who are late for work more than 5 times per month will be sent a warning letter by the system. This letter is auto generated.

➤ **Home**

To return back to the home page, which is the welcome page.

➤ **Log Out**

To log out from the whole system.

### 3.5.1.2 Functional Requirements for Administrator Module

➤ **Attendance Setting**

Administrator can find the attendance records of particular staffs or all staffs on particular date. The attendance record can be updated, deleted and a new attendance record can even be added into the database. There are functions provided in case of the power failure. Administrator can clock in, clock out or update status of all the staffs to normal working hours.

➤ **Working Hours Setup**

A table of working hour's setup will be displayed as administrator clicks on this link. The setting can be updated or deleted. Administrator has the right to add new working hour into the database too.



### ➤ Reports

Like the user module, administrators can view four types of reports, which are mentioned above. Authority and responsibility has been delegated to administrator to generate the report and view the reports for all or each staff. There are many selections provided to administrator for the ease of generating reports, which are date, name and status. The reports can be sorted by name, status or date. Beside that, administrator can either view the details of the whole reports or the summary reports.

### ➤ Hot site browsing and adding

The function is same like what has been described in the user module. The different is the administrator has the right to add the new URL to the system.

## 3.5.2 Non-Functional Requirements

Following are the non-functional requirements of the system:

### ➤ Reliability

- Reliability is the extents to which a system can be expected to perform its function with required precision and accuracy. Thus, the system should be reliable in performing its daily functions and operations. For example, whenever a button is clicked, the system should be able to perform some functionality or generate some message to inform the user what is happening.

### ➤ Scalability

- Due to the web based and distributed nature of the project's implementation; the scalability issue can be addressed rather well by separating the key modules. Each module should has the capability to run on separate machines that can be expanded or contracted as well as run from within a single machine. Database scalability issues should be resolved using distributed database architecture. The AMS web-based application should have the scaling to increase bandwidth or adding web servers.



### ➤ Security

- The system should be equipped with sufficient security. Each access by the user should be authenticated and validated by the system. The system should not show any potential of leakage of information. The password should be encrypted.

### ➤ Database backup

- The system should be able to restore to its normal operation from any potential disaster such as power failure, sabotage and natural disaster. The continuous of operation should be ensured so that to mitigate the losses of faculty. Other than that, the performance and the space of database should be tuned. For an example, providing the daily and yearly attendance table so that all the data will be destroyed after a certain period, which meet the certain criteria.

### ➤ Inter-Operability

- The system should have the capability to work with different types of applications to share data and process. The system can be integrated with Leave Management System and other GOE system. When the user login to the system, he or she has the option to either access to AMS or LMS.

### ➤ Flexibility

- The system should have the capability to take advantage of new technologies and resources. For an example, it is flexible to use the stored procedure in replacing the Transact SQL statements. Beside that, it should be able to be implemented when the system evolves or there is a change in environment.



### 3.6 Development Tools

An analysis has been carried out on current development tools in section 2.2.3.2. From the analysis, most of the tools are stable, suitable and able to support among each other. They are unlikely to change since they have been implemented. In this project, the main concerns are the language being used and its database server. Since the benefits gained are more than if they are changed to other options; and the process of enhancement would not be restricted, therefore, the system should continue using the existing developments tools. Below are the technologies and tools that will be remained in this project:

#### ➤ Development Technologies

- Microsoft Windows NT Server 4.0 as the development platform
- Internet Information Server 4.0 (IIS) as the web server
- Microsoft SQL Server 7.0 as the database server
- Internet Explorer 5.0 as the browse

#### ➤ Development Software

- Visual Interdev 6.0
- Visual Basic 6.0
- Adobe Photoshop 5.0

#### ➤ Programming Language

- Active Server Pages (ASP)
- Active X DLL and Components
- Hypertext Markup Language (HTML)
- Structure Query Language (SQL)
- VB Script



### 3.6.1 Reason of Remaining ASP Technology

- It can support the integration of AMS and LMS and customize the web page to the specific need of each individual.
- It works together with Window NT and IIS to provide a comprehensive set of technologies that enable secure exchange of information over network.
- It is able to create pages that are sensitive to factors such as time and place, and the users' identity and previous choices and actions.
- ASP coding can be embedded with SQL.

### 3.6.2 Reason of Remaining SQL Database Server

- It is integrated with the security system in Window NT.
- It has the functionality of stored procedures. The advantages of using the store procedures are:
  - It is quicker to run a stored procedure because it is stored in a precompiled form of the database. This means that, unlike the Transact SQL statements that we write in the Active Server Page (ASP) page, they don't need to be compiled before they can be executed.
  - It makes code more readable.
  - It can be used over and over again in ASP pages.
- It can perform the scheduled jobs and triggers such as generating auto mails to those who are late for work and update daily attendance record automatically. Other than that, it is able to perform the archive function to control the growth of database.

### 3.7 Conclusion

The requirements of the system are elicited from the studies and analysis done during the literature review. With the development model and tools that have been determined, the new features and enhancement should overcome all the problems and weaknesses of the current AMS as well as increasing its effectiveness and efficiency.



## CHAPTER 4 : SYSTEM DESIGN

With the requirements that have been identified in chapter 3, the system would be designed to transform the informal idea to detailed implementation descriptions. Other than that, it will show how the integration of the entire system will be conducted. The system design of AMS would be carried out in few methods. They are:

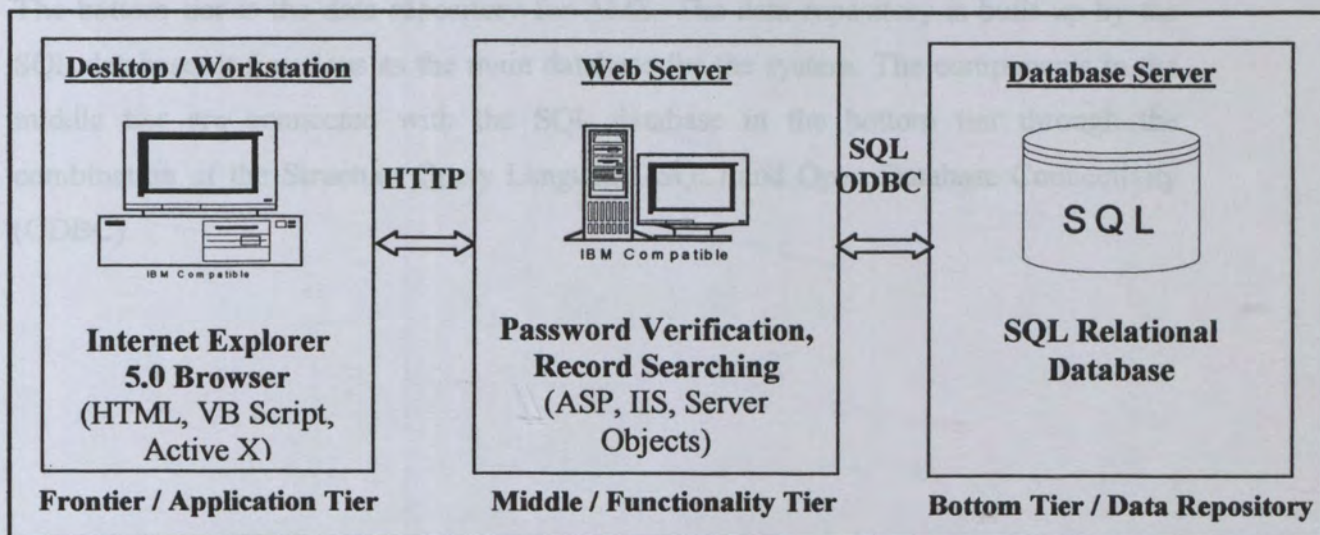
- System Functional Design
- Network Design
- Database Design
- Graphical User Interface Design

### 4.1 System Functional Design

Under the system Functional Design, the system is designed from a functional viewpoint, starting with high-level view and progressively refining this into a more detailed design. System architecture, data flow diagram and system chart for AMS will be examined and refined in this stage.

#### 4.1.1 System Architecture

As the enhancement project, AMS will continue using its existing architecture, which is the **Three-Tier Client / Server Architecture** as showed in the Figure 4-1.



**Figure 4-1 AMS Three-Tier Client / Server Architecture**



### **Front Tier / Application Tier**

The frontier is the application tier consists of all the necessary applications. In this layer, the main application component that appears to the users under the AMS is the Internet Explorer 5.0 browser. This layer will provide the user interface. HTML, VB Script and Active X control are used to activate the application layer. All of them provide the most flexible and dynamic interface for the users. The application is always reside within the web server, which is the Internet Information Server 4.0.

### **Middle / Functionality Tier**

The middle tier is known as the functionality or service tier. The communications between this tier and the frontier depends on the Hypertext Transfer Protocol (HTTP) for the web pages transfer. The functionality tier consists of the components that are required to support the Attendance Management System such as password verification, searching for records and other configuration. All these components require Active Server Pages and the ASP server objects to perform the functions in the web servers. The IIS in this tier will processes the request from the client and produces the result in web pages format. The IIS will also process any data request by the user through the linking to the database server, which is resided in the bottom tier. Additional activities will be performed by IIS during the data processing.

### **Bottom Tier / Data Repository**

The bottom tier is the data repository for AMS. The data repository is built up by the SQL database. It functions as the main database for the system. The components in the middle tier are connected with the SQL database in the bottom tier through the combination of the Structure Query Language (SQL) and Open Database Connectivity (ODBC).



## 4.1.2 Data Flow Diagram

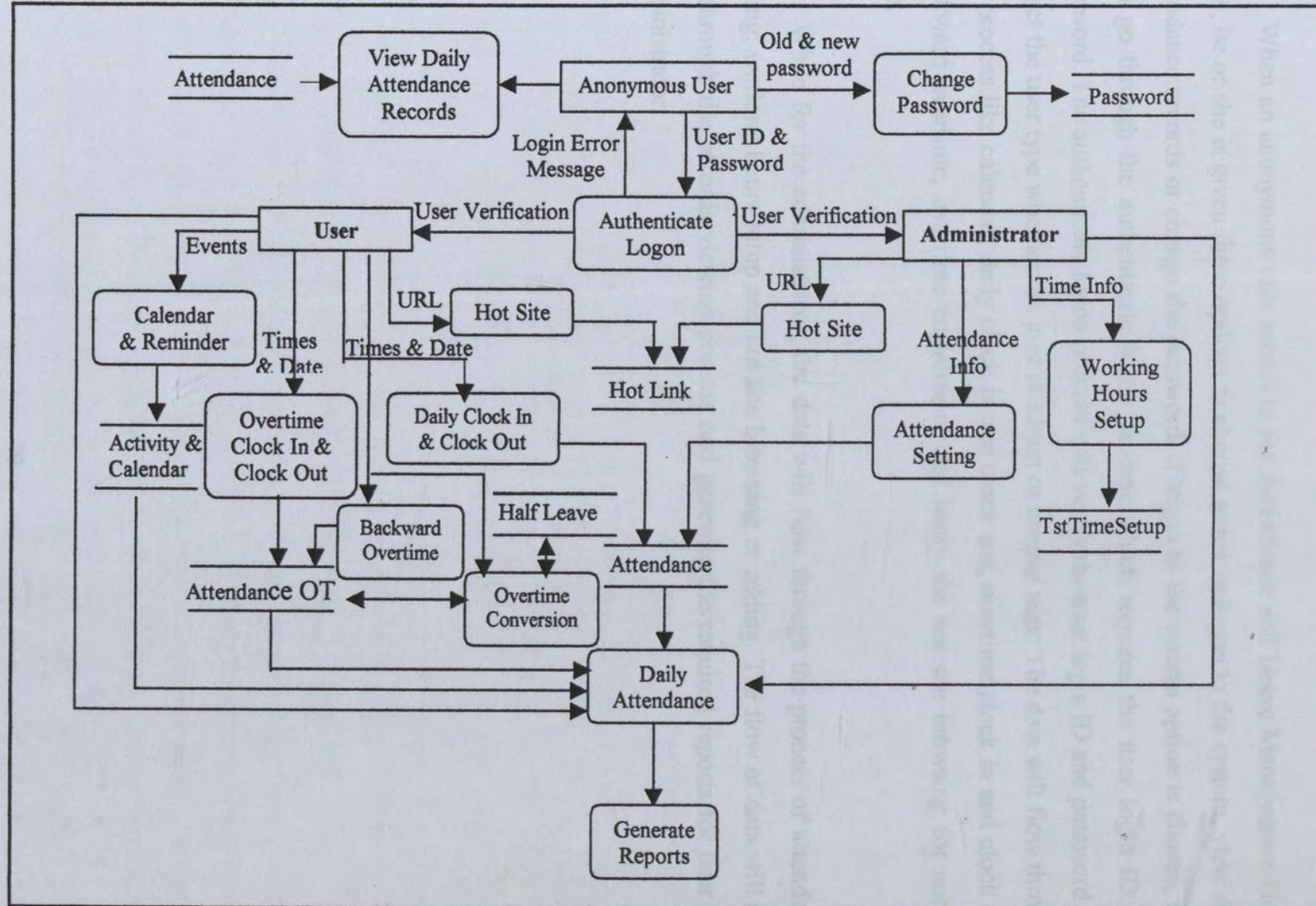


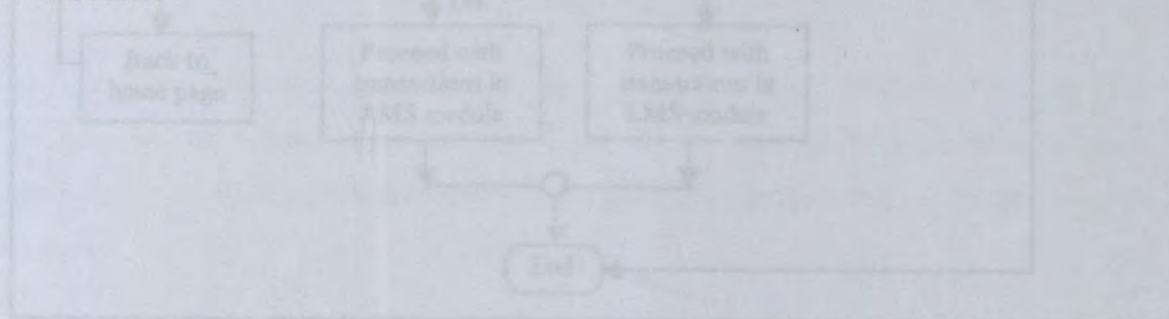
Figure 4-2 AMS Data Flow Diagram



Data-flow models are an intuitive way of showing how data is processed by a system. It shows how data flows through a sequence of processing steps. There are two main modules in the AMS system, which are user and administrator modules. Figure 4-2 shows the data-flow diagram of this system.

When an anonymous user access to the Attendance and Leave Management Home page, he or she is given three options to choose either to logon to the system, view daily attendance records or change the password. If logon to the system option is chosen, user will go through the authenticate logon process which requires the user login ID and password. The authenticate logon process will verify the user login ID and password and detect the user type whether the user is admin or normal user. The data will flow through the process like calendar, daily clock in and clock out, overtime clock in and clock out, backward overtime, overtime conversion, and lastly the hot site browsing for normal user.

While for the administrator, the data will flow through the process of attendance setting; working hours setup and hot site browsing or adding. The flow of data will also go through the records viewing process and generate the required reports for user and administrator.

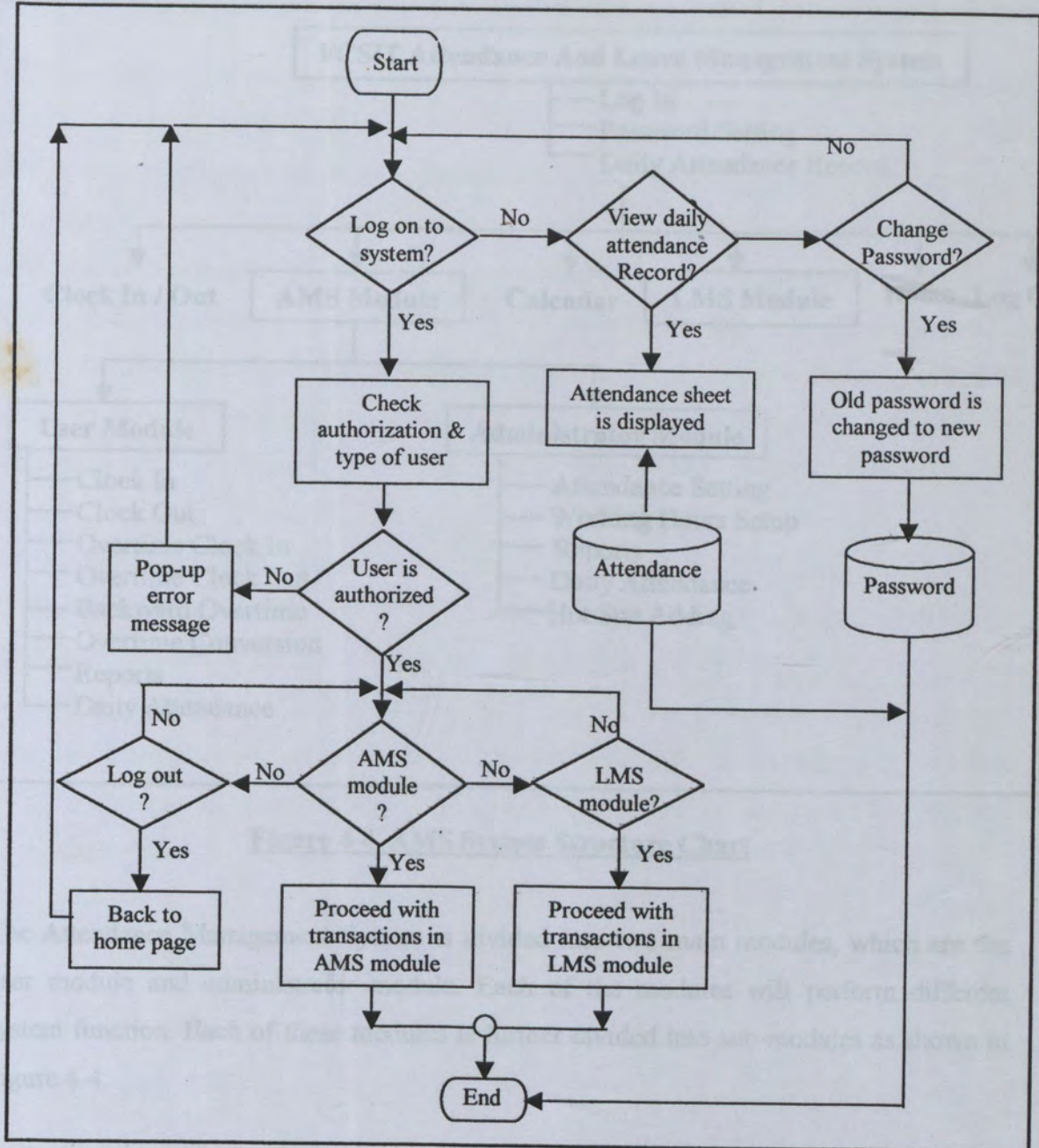


**Figure 4-3 AMS & LMS Integrated Data Process Flow**

AMS and LMS are enhanced to have a better integrated interface. Users no need to enter password twice to access to both applications. The daily attendance records can be viewed without login to the system. This means the students can also access to the attendance sheets of all the staffs in faculty. The proposed data flow for the integrated interface is shown in Figure 4-3.



### 4.1.3 Process Flow for AMS and LMS Integration

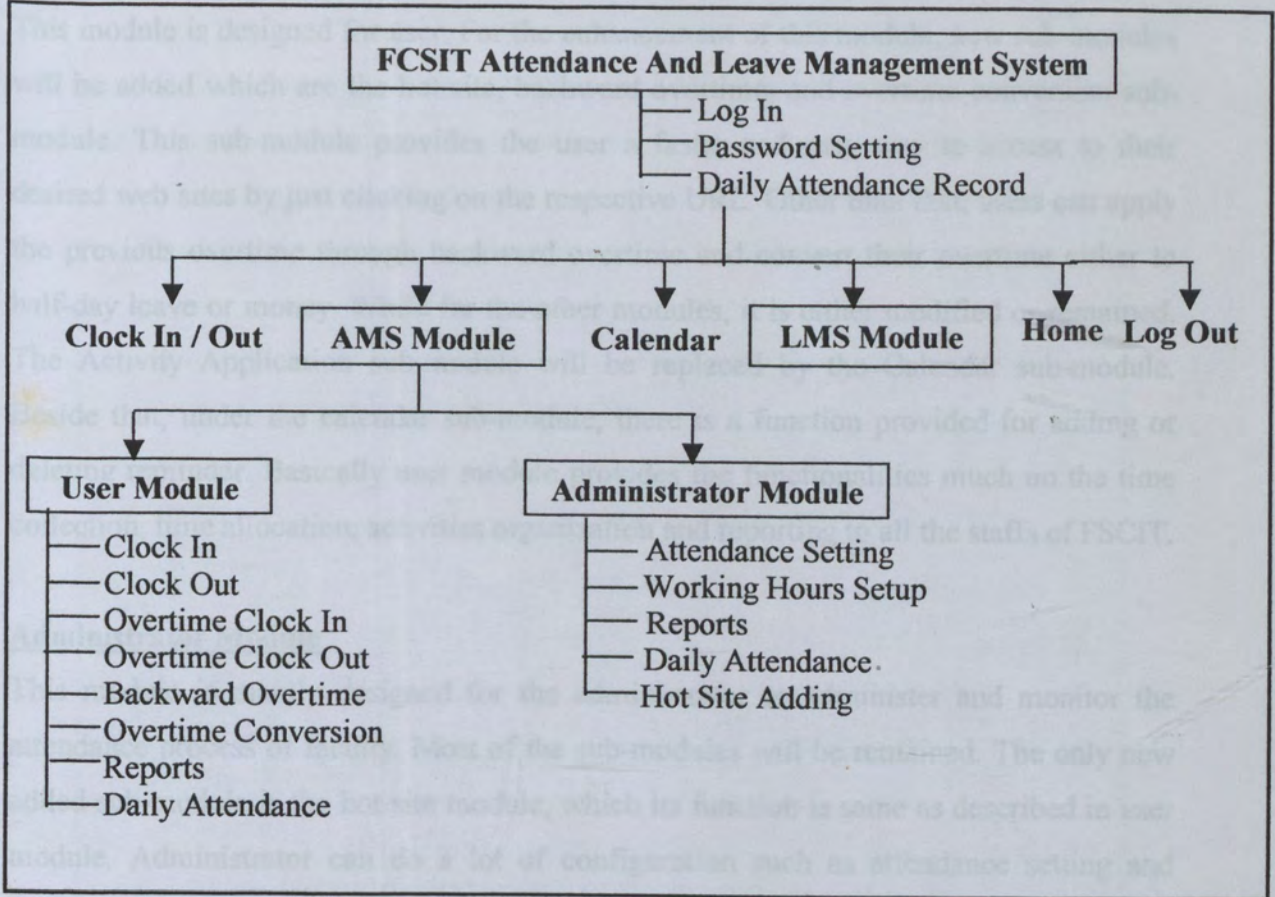


**Figure 4-3 AMS & LMS Integration Data Process Flow**

AMS and LMS are enhanced to have a better-integrated interface. Users no need to enter password twice to access to both applications. The daily attendance records can be viewed without logon to the system. This mean the students can also access to the attendance sheets of all the staffs in faculty. The proposed data flow for the integrated interface is shown in Figure 4-3.



#### 4.1.4 System Structure Chart



**Figure 4-4 AMS System Structure Chart**

The Attendance Management System is divided into two main modules, which are the user module and administrator module. Each of the modules will perform different system function. Each of these modules is further divided into sub-modules as shown in Figure 4-4.



### **User Module**

This module is designed for user. For the enhancement of this module, new sub-modules will be added which are the hot site, backward overtime, and overtime conversion sub-module. This sub-module provides the user a faster and easy way to access to their desired web sites by just clicking on the respective URL. Other than that, users can apply the previous overtime through backward overtime and convert their overtime either to half-day leave or money. While for the other modules, it is either modified or remained. The Activity Application sub-module will be replaced by the Calendar sub-module. Beside that, under the calendar sub-module, there is a function provided for adding or deleting reminder. Basically user module provides the functionalities much on the time collection, time allocation, activities organization and reporting to all the staffs of FSCIT.

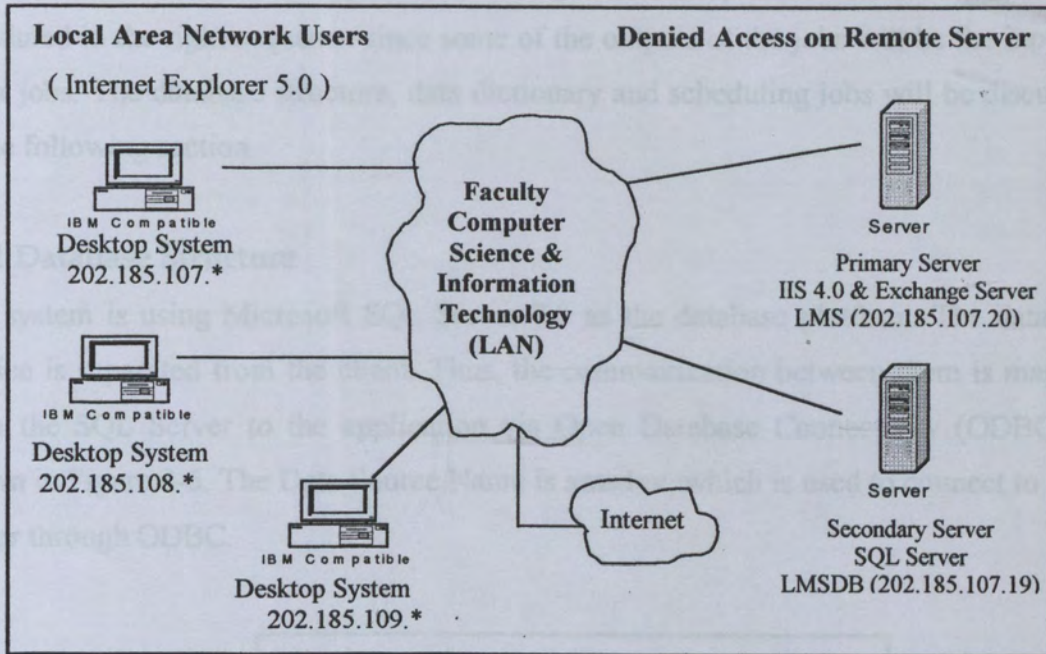
### **Administrator Module**

This module is mainly designed for the administrator to administer and monitor the attendance process of faculty. Most of the sub-modules will be remained. The only new added sub-module is the hot site module, which its function is same as described in user module. Administrator can do a lot of configuration such as attendance setting and working hours setup. The most important thing is it provides the customized and dynamic reporting for analysis purpose.



## 4.2 Network Design

The network design is very important. The current network design of Attendance Management System will be carried on. This system only allows local area network users to access the application. The remote users are unable to access the above application. The connection between the user desktop and the web server is linked with the Internet facilities. The entire connection is communicated with TCP/IP protocol.



**Figure 4-5 AMS Network Setup Design**

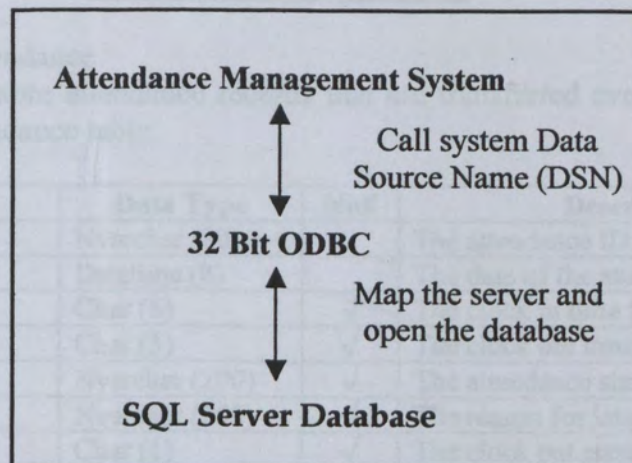


### 4.3 Database Design

The database designs will inevitable affect all the modules in Leave Management System and within Attendance Management System itself. Thus, under this stage the data models that were developed will be translated into data structure. For the new proposed features, new tables will be created and their relationship will be defined and examined to avoid the inconsistency and redundancy of data. Beside that, cursors have been design to be run as schedule jobs to maintaining and monitoring AMS. The jobs need to be designed and scheduled in the right sequence since some of the outputs of the jobs will be the input of other jobs. The database structure, data dictionary and scheduling jobs will be discussed in the following section.

#### 4.3.1 Database Structure

The system is using Microsoft SQL Server 7.0 as the database platform. The database service is separated from the client. Thus, the communication between them is mapped from the SQL Server to the application via Open Database Connectivity (ODBC) as shown in Figure 4-6. The Data Source Name is **amsdev**, which is used to connect to SQL server through ODBC.



**Figure 4-6 Database Structure**



### 4.3.2 Data Dictionary

The current database name for this system is **LAMS**. **LAMS** is the abbreviation of Leave and Attendance Management System. Below are the new tables and modification of some of the current tables.

\*Represent the **primary key**

- 1) Current **Attendance** table will be separated into two new tables to perform the archive function. They are **Attendance** and **Yearly Attendance**.

**Name :** Attendance

**Description :** To store daily attendance records

Field Name	Data Type	Null	Description
TranID	Nvarchar (30)		The attendance ID for the record
*Date	Datetime (8)		The date of the attendance
Clock In	Char (5)	✓	The clock in time for user
Clock Out	Char (5)	✓	The clock out time for user
Status	Nvarchar (200)	✓	The attendance status for user
Remark	Nvarchar (255)	✓	The reason for late
Logout	Char (1)	✓	The clock out status
Confirmation	Bit (1)		The update status
Ustatus	Bit (1)		The user status
IPAdd	Nvarchar (20)	✓	The IP address of the client PC
*ApplicantID	Int (4)		The user ID

**Table 4-1 Attendance Table**

**Name :** Yearly Attendance

**Description :** To store attendance records that are transferred everyday from daily attendance table.

Field Name	Data Type	Null	Description
TranID	Nvarchar (30)		The attendance ID for the record
*Date	Datetime (8)		The date of the attendance
Clock In	Char (5)	✓	The clock in time for user
Clock Out	Char (5)	✓	The clock out time for user
Status	Nvarchar (200)	✓	The attendance status for user
Remark	Nvarchar (255)	✓	The reason for late
Logout	Char (1)	✓	The clock out status
Confirmation	Bit (1)		The update status
Ustatus	Bit (1)		The user status
IPAdd	Nvarchar (20)	✓	The IP address of the client PC
*ApplicantID	Int (4)		The user ID

**Table 4-2 Yearly Attendance Table**



- 2) As the Hot Site browsing and adding feature is added, **HotLink** is a new table that would be created in database.

**Name :** HotLink

**Description :** To store hot sites' URL

Field Name	Data Type	Null	Description
*URLAdd	Nvarchar (100)		The URL address of the hot site
Name	Nvarchar (100)		The name of the hot site
IntType	Char (10)		The type of the web site
ID	Decimal (9)		The ID for web site

**Table 4-3 HotLink Table**

- 3) The calendar sub-system will create a new table called **calendar**. It acts as diary for users.

**Name :** Calendar

**Description :** calendar for users

Field Name	Data Type	Null	Description
*TransID	Int (4)		The calendar transaction ID
Dte	Datetime(8)		The date of calendar
Text field	nvarchar (200)	√	The details of activity / events
ApplicantID	Int (4)		The applicant ID

**Table 4-4 Calendar Table**

- 4) The **activity** table in database was modified to acts as the reminder table.

**Name :** Activity

**Description :** To record all activities/events which act as reminder for users

Field Name	Data Type	Null	Description
TransID	Decimal (9)		The reminder record ID
*Date	Datetime(8)		The date of activity
*Start	Datetime(8)		The start time of the activity
Ended	Datetime(8)	√	The end time of the activity
Remark	Nvarchar (255)		The remark of activity
Location	Nvarchar (100)		The location of the activity
*ApplicantID	Int (4)		The applicant ID
usage	Int (4)		The reminder is for 1-personal, 2-public, 3-both.

**Table 4-5 Activity Table**



- 5) The **HalfLeave** table in database will be modified as the overtime to half-day leave conversion needs to be approved and the unrelated fields have been taken out.

**Name :** HalfLeave

**Description :** To record all transactions of overtime to half-day leave conversion.

Field Name	Data Type	Null	Description
TransID	Decimal (9)		The transaction ID of half day leave conversion
*ApplicantID	Int (4)		The applicant ID
*Date	Datetime(8)		The date of half-day leave applied
TypeLeave	Nvarchar(50)		The type of half-day leave applied
Reason	Nvarchar(100)		The reason of applying half-day leave
OTuse	Int (4)		The amount of OT converted to half-day leave
Approved	Int(4)		1 – approved by first approver 3 – approved by second approver 4 – application is rejected
ActionBy	Nvarchar (50)	√	The name of the approver
Action Date	datetime (8)	√	The date approved by approver
Processed	Int (4)		0 –the application has not been processed 1 –the application has been processed by first approver 2–the application has not been processed by second approver
Revoking	Int (4)		0 – the application is not revoked 1 – the application is revoked
RApproved	Int (4)		Same with Approved but for revocation
RProcessed	Int (4)		Same with Processed but for RProcessed
NoDay	Int (4)		Number of day applied for half-day leave

**Table 4-6 HalfLeave Table**

- 6) The **Cmoney** table is added into database as overtime is allowed to be converted into cash

**Name :** Cmoney

**Description :** To record all transactions of overtime to money conversion.

Field Name	Data Type	Null	Description
*TransID	Decimal (9)		The transaction ID of money conversion
ApplicantID	Int (4)		The applicant ID
Date	Datetime(8)		The date of OT to money convesion
OTuse	Int (4)		The amount of OT converted to money
Approved	Int(4)		1 – approved by first approver 2 – approved by second approver 3 – application is rejected
ActionBy	Nvarchar (50)	√	The name of the approver



Action Date	datetime (8)	√	The date approved by approver
Processed	Int (4)		0 –the application has not been processed 1 –the application has been processed by first approver 2–the application has not been processed by second approver
Revoking	Int (4)		0 – the application is not revoked 1 – the application is revoked
RApproved	Int (4)		Same with Approved but for revocation
RProcessed	Int (4)		Same with Processed but for RProcessed
Remark	Nvarchar (100)	√	Remark for converting OT to money

**Table 4-7 Cmoney Table**

- 7) Current **AttendanceOT** table was modified to meet the new requirement where all the taken overtime must be approved.

**Name :** AttendanceOT

**Description :** To record all transactions of overtime.

Field Name	Data Type	Null	Description
*TranID	Nvarchar (30)		The transaction ID of overtime
*Date	Datetime (8)		The date of the overtime
Clock In	Char (5)	√	The clock in time for overtime
Clock Out	Char (5)	√	The clock out time for overtime
Status	Nvarchar (20)		The overtime status for user
Remark	Nvarchar (255)	√	The reason for overtime
Logout	Char (1)	√	The overtime clock out status
Confirmation	Bit (1)		0 – the overtime is not confirm 1 – the overtime is confirm (clock in & clock Out)
Ustatus	Bit (1)		0 – for academic staff 1 – for non-academic staff
IPAdd	Nvarchar (20)	√	The IP address of the client PC
*ApplicantID	Int (4)		The user ID
Approved	Int(4)		1 – approved by first approver 2 – approved by second approver 3 – application is rejected
ActionBy	Nvarchar (150)	√	The name of the approver
Action Date	datetime (8)	√	The date approved by approver
Processed	Int (4)		0–the application has not been processed 1–the application has been processed by first approver 2–the application has been processed by second approver
NoTime	Int (4)		The amount of time taken for OT

**Table 4-8 AttendanceOT Table**



- 8) The **TstTimeSetUp** table is modified to add in new type of working hours, such as start working hours and end working hours of half-day leave.

**Name :** TstTimeSetUp

**Description :** To record different type of working hours.

Field Name	Data Type	Null	Description
*TypeID	Int (4)		The type ID for working hours
*Day	Char (10)		The day's name
Starttime	Datetime(8)		Start time of working day
Endtime	Datetime (8)		End time of working day
Extra	Smallint (2)		Plus and minus basis
DayID	Int (4)		The ID of day
*TypeName	Nvarchar (50)		The type of working hours

**Table 4-9 TstTimeSetUp Table**

### 4.3.3 Jobs Scheduling

Jobs that have been identified in **section 3.2 and 3.3** in the chapter of system analysis and requirement are scheduled at different interval time in a right sequence. For an example, jobs such as Attendance Sheet, Detect Leave, and Detect Half-Day Leave are scheduled before the Remind job. This is because only after creating the list of name for attendance sheet, detecting who are taking leave and half-day leave, the Remind job just can be triggered to remind those who have not clocked in after 12:00 pm. Another example is the Archive Attendance job, which can only be done after Update Status job.

Below are the jobs that have been scheduled in SQL Server Agent:

#### ➤ Archive Attendance

The daily attendance records will be archived / transferred to Yearly Attendance table everyday at 2:00 am.

#### ➤ Archive Yearly Attendance

All the good attendance records with the status 'Present' and remark 'Null' will be deleted from Yearly Attendance table. This job is scheduled to be run every 12 months on the 1<sup>st</sup> Monday of that month at 12:00 am.



### ➤ Attendance Sheet

This job will create records of all the staffs in attendance table everyday at 5:30 am. If the particular day were a holiday or Sunday, the records would not be created. The purpose of this job is creating attendance sheet for viewing. When the staff is coming to work and clock in, his or her record will then be updated. If the server is down and the job is failed to execute, all records will then be created through the system application when the first staff comes to clock in.

### ➤ Detect Leave

Everyday, who is taking leave will be detected at 5:40 am. This job will then update the corresponding record into daily attendance table, indicating the person is on-leave.

### ➤ Detect Half-Day Leave

If anyone is taking half-day leave, this job will update the corresponding records into daily attendance table. It is scheduled to run every 7 hours between 5:55 am and 5:55 pm. The first running is at 5:55 am to detect those who are taking morning half-day leave. After 7 hours, which is around 12:55 pm, it will begin to detect those who are taking afternoon leave.

### ➤ Remind

If there is anyone forgets to clock in after 12:00 pm on a normal working day, a mail will be sent to remind him/her. This job will occur everyday at 12:10 pm.

### ➤ Late Comer

If there is anyone late for work more than 5 times in a month, a warning mail will be sent to him/her.



### ➤ Update Status

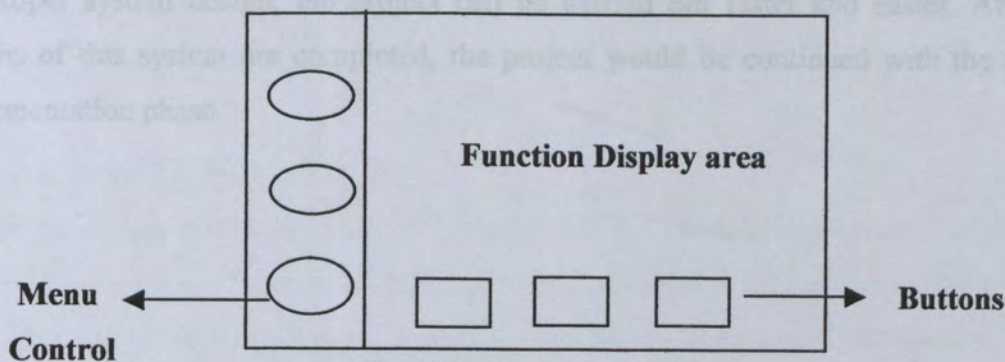
All records in attendance table will be updated everyday at 1:30 am before transferring to yearly attendance table. If a person neither clock in or clock out, the status will be updated as 'Absent'. If he/she has clocked in but forgets to clock out, the logout status will be updated as 'Y'. In the other hand, the logout status will be updates as 'N'. Then, if the confirmation field is still '0', it will be updated to '1'.

## 4.4 Graphical User Interface Design

The current interface design of AMS was improved and was user-centered. It is logical and consistent and help users recover from errors. Users no need to remember any dos command and what they need to do is just some mouse clicks. Since some of the graphical displays of the current system fail to present trends and approximate values, they have been changed. Colour used for this application now is sparingly and consistently. This good design can minimize the likelihood of errors occurred.

### 4.4.1 Design of Screen

The screen design of AMS is divided into two, which are user screen and administrator screen. However, all screen designs will be standard and consistent by using the frames. The left frame contains the menu control while the right frame is the functional display area. Button will be displayed at the appropriate area for user to click. Below is the example of screen design:



**Figure 4-7 AMS Interface Design**



### 4.4.2 Design of Output

The design of output is important to serve the purpose of providing the systematic as well as accurate data to users. In AMS, users are provided options to generate reports according to their wish. The layout of reports is designed to look and feel the same way to the user. Below is one of the examples of the design of daily attendance report:

Faculty Computer Science And Information Technology  
University Malaya

Attendance Record Date : Wednesday, November 01, 2000

List of Normal Attendance

#### Academic Staff

Full Name	Attendance Status	Remark
Abdullah Gani	On-Leave	Rest
Abrizah Abdullah	Present	
Ali Fauzi Ahmad Khan	Present	
Chiew Thiam Kian	Present	
Diljit Singh (PM. Dr.)	On-Leave	Rest
Fariza Hanum Md. Nasaruddin	Present	
Hannyzzura Pal @ Affal	Present	

**Figure 4-8 AMS Output Design**

### 4.5 Conclusion

The system designs require experience as well as systematic planning because only with the proper system design, the project can be carried out faster and easier. After the designs of this system are completed, the project would be continued with the system implementation phase.



## CHAPTER 5 : SYSTEM IMPLEMENTATION

The system implementation of AMS will be divided into two components, which are the platform development and the modules implementation.

### 5.1 Platform Development

Before AMS is implemented, the implementation environment must be established. The initial development of platform involves setting up the operating system, which is *Window NT server 4.0*. Since AMS is developed under three-tier architecture, the implementation will then be continued by setting up web server and database server. The web server of AMS is *Microsoft Internet Information Server* while the database server is *Microsoft SQL server 7.0*. The web server will be the Primary Domain Controller (PDC) and the database server will be the Backup Domain Controller (BDC). BDC can be setup during the installation of Window NT or by using Window NT Server Manager in the PDC. Lastly, mail server, which is *Microsoft Exchange Server 5.5* is setup too for the mailing purposes. Below is the summary of establishing the implementation environment:

- Install Windows NT 4.0
- Install the NT Option Pack 4 (Which includes the Service Pack 3, Internet Information Server 4.0 and Microsoft Transaction Server)
- Install NT Service Pack 6
- Install SQL Server 7.0
- Install Microsoft Exchange Server 5.5 from Microsoft Back Office

#### 5.1.1 Window NT Server 4.0

Before the installation of Window NT Server 4.0, the hard disk is formatted using NT File system format to ensure a more stable and secured NT transaction across the platform. The partition of hard disk can be created too prior to the installation. This is done for the purpose of more systematic management of the computer storage.



During the installation of Window NT Server 4.0, all the instructions must be properly followed. For an example is setting up the domain name of AMS, which is LAMS. The Internet Explorer 2.0 will come along with the installation of Window NT Server 4.0. After the installation of Window NT Server, the machine is rebooted. Then, the Internet Explorer 2.0 is upgraded by installing the latest version, which is Internet Explorer 5.0.

### 5.1.2 Internet Information Server

After installing the IIS, a virtual directory has to be created to enable user access to the application. The virtually directory is pointed to the physical directory where all the system files are found. The default page for the Attendance and Leave Management System is '**frontframe.html**.' For the AMS users, they can access the application through internet and intranet. However, two files in the virtual directory are configured where they only grant access to the local area network, which the IP addresses start from **202.185.107.1** until **202.185.109.254**. The two files are '**in.asp**' and '**out.asp**'. The application can be accessed through the following address <http://lms/>

### 5.1.3 Microsoft SQL Server 7.0

The SQL server is installed in another Compaq server. It is separated from the IIS and Exchange Server. Upon successful installation, a database named **LAMS** is created. Then, it is followed by creating all tables of the system. Other than that, bigger hard disk is allocated for the database to maximize the performance of the SQL server and to ensure there is enough space to store all the data. The file growth of the database is set to 10% of the original database.

All the data in the existing database (Goelas) is transferred into this new database. In order to map the database to the web server, SQL Data Source Name (DSN) needs to be configured. This is done through the ODBC in the web server machine. The **System Data Source Name** is '**amsdev**'. The ODBC could map to the SQL server even it is in the different machine. This is because the web server could open the database storage by



just calling the system DSN. Finally, the system SQL server agent and SQL mail are also configured to perform scheduled task and mailing.

### SQL Server Agent

SQL Server Agent is an auxiliary operator inside the **management folder** of server group. It is responsible for handling the repetitive tasks and exception handling conditions, which is:

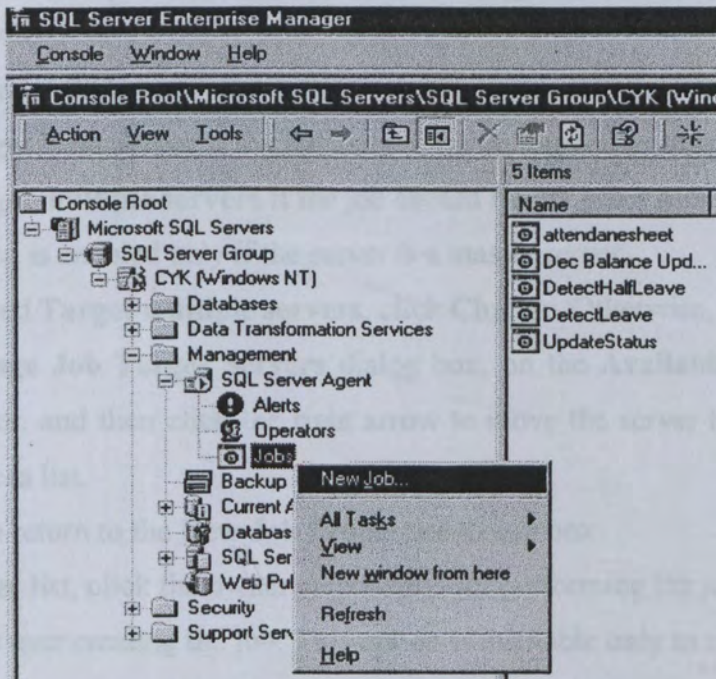
- Running SQL Server Task that scheduled to occur at specific times or intervals.
- Detecting specific conditions for which administrators have defined an action, such as alerting someone through pages or e-mail, or a task that will address the condition.

For AMS, an operator is needed to act as the system administrator. He is the one to be informed if any scheduled job failed (or succeed, depend on the setting). Below is an example of how to create the jobs as scheduled in **section 4.3** in SQL Agent.

### To Create a Job for ArchiveAttendance

1. Right-click Jobs on SQL Sever Agent root; then click **New Jobs** as shown in

Figure 5-1.



**Figure 5-1 SQL Server Agent Root**



2. In the **New Job Properties** dialog box, on the **General** tab, in **Name**, enter a name for the job. The maximum number of characters is 128.

**New Job Properties - GSE**

General | Steps | Schedules | Notifications

**Name:** ArchiveAttendance **Source:** (local)

**Created:** (Not yet created) ☒ **Enabled** ☐ **Target local server** ☐ **Target multiple servers:**

**Category:** [Uncategorized (Local)] **Change**

**Owner:** sa

**Description:**

**Last modified:** (Not applicable) **Change**

OK Cancel Apply Help

**Figure 5-2 Enter Name for New Job**

3. If you do not want the job to be able to be run immediately following its creation, clear **Enabled**. For example, if you want to test a job before it is scheduled to run, disable the job.
4. Under **Source**, select:
  - **Target local server** if the job should run on this server only.
  - **Target multiple servers** if the job should run on other servers. This option is enabled only if the server is a master server.
5. If you selected **Target multiple servers**, click **Change**. Otherwise, skip to Step 7.
6. In the **Change Job Target Servers** dialog box, on the **Available Servers** tab, click a server, and then click the right arrow to move the server to the **Selected target servers** list.
7. Click **OK** to return to the **New Job Properties** dialog box.
8. In the **Owner** list, click the owner responsible for performing the job. The default owner is the user creating the job. This option is available only to members of the **sysadmin** role.



9. In the **Description** box, enter a description of what the job does. The maximum number of characters is 512.
10. Click the **Steps** tab; then click **New**. Each job must have at least one step. A step must be created with the job before the job can be saved.

The screenshot shows a dialog box titled "New Job Properties - GSE" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Steps", "Schedules", and "Notifications". The "Steps" tab is currently selected. Inside the dialog, there is a table with the following columns: "ID", "Step Name", "Type", "On success", and "On failure". The table is currently empty. Below the table, there are several controls: "Move step:" with up and down arrow buttons, "Start step:" with a dropdown menu, and a row of buttons: "New...", "Insert...", "Edit...", and "Delete". At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

ID	Step Name	Type	On success	On failure
----	-----------	------	------------	------------

**Figure 5-3 Before Step of New Job is Created**

11. In the **General** tab, type in the **Step Name**. Choose the **type** of job to be created from the list of combo box as well as the **database** name where the job will be put in. The type of job for ArchiveAttendance is Transact-SQL Script (TSQL) while the database is LAMS. For **Command**, click **Open** to browse the ArchiveAttendnace code / cursor that has been created as shown in Figure 5-4.



**New Job Step - GSE**

General | Advanced

Step name: ArchiveAttendance

Type: Transact-SQL Script (TSQL)

Database: LAMS

Command:

```
Select (*) from LAMS.Attendance
if
    0
Declare @TransID decimal,
        @Date datetime,
        @ClockIn char(5),
        @ClockOut char(5);
```

Open... Parse

Goto Step: Next Previous OK Cancel Apply Help

**Figure 5-4 Putting Code into Command**

12. After that, click **OK**. The step is created and it will return back to the job properties. The screen is shown as below.

**New Job Properties - GSE**

General | Steps | Schedules | Notifications

ID	Step Name	Type	On success	On failure
1*	ArchiveAttendance	Transact-SQ...	Goto next step	Quit with failure

Move step: [Up] [Down] Start step: (1) ArchiveAttendan... New... Insert... Edit... Delete

OK Cancel Apply Help

**Figure 5-5 After Step of New Job is Created**



13. Then, move on to the **Schedules** tab and click **New Schedule**. The screen is shown as in Figure 5-6. In the **Name** box, enter a name for the new schedule. The maximum number of characters is 128.
14. Clear **Enabled** if you do not want the schedule to take effect immediately following the creation.

**New Job Schedule - GSE**

Name:  ☒ **Enabled**

**Schedule Type**

☐ Start automatically when SQL Server Agent starts

☐ Start whenever the CPU(s) become idle

☐ **One time** On date:  At time:

☒ **Recurring**

Occurs every 1 week(s) on Sunday, at 12:00:00 AM.

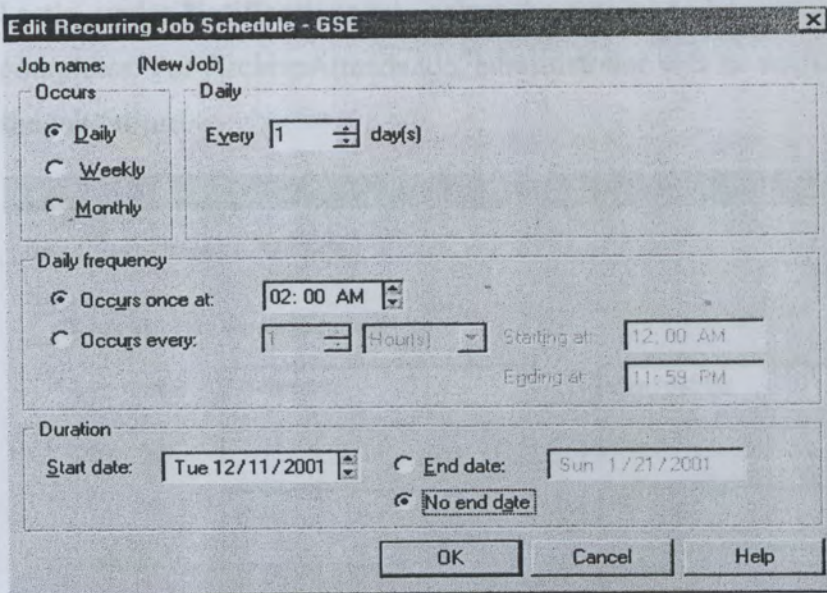
**Figure 5-6 Schedule the New Job**

Under **Schedule type**, select the schedule type that meets your needs:

- If you selected **One time**, enter values in the **On Date** and **At time** boxes.
- If you selected **Recurring**, click **Change**, and then complete the **Edit Recurring Job Schedule** dialog box.

15. Since ArchiveAttendance needs to be recurred at 2:00am everyday, so the button **Change** is clicked. A screen will be displayed as shown in Figure 5-7.





**Edit Recurring Job Schedule - GSE**

Job name: [New Job]

Occurs: ☒ Daily ☐ Weekly ☐ Monthly

Every 1 day(s)

Daily frequency:

☒ Occurs once at: 02:00 AM

☐ Occurs every: 1 Hour(s) Starting at: 12:00 AM Ending at: 11:59 PM

Duration:

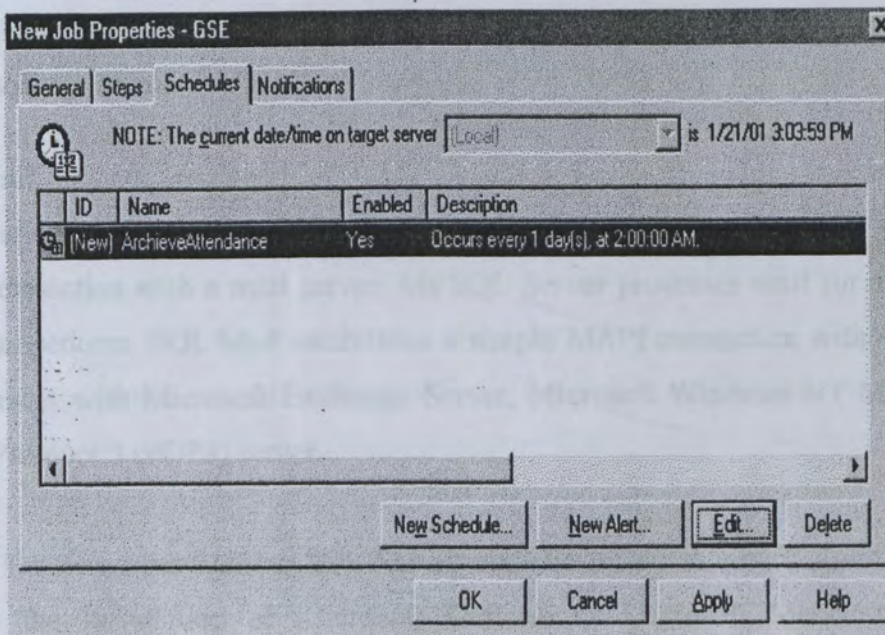
☒ Start date: Tue 12/11/2001 ☐ End date: Sun 1/21/2001

☒ No end date

OK Cancel Help

**Figure 5-7 Recurring Job Schedule**

16. Under **Occur**, select daily and every 1 day(s) under daily frame. For daily frequency, select occurs once at 2:00am. Under duration, select the desired start date and end date. For ArchiveAttendance, it will be started at 12/11/2000 and there is no end date. After every single item has been selected, click OK. The screen will be depicted as below.



**New Job Properties - GSE**

General | Steps | Schedules | Notifications

NOTE: The current date/time on target server [Local] is 1/21/01 3:03:59 PM

ID	Name	Enabled	Description
(New)	ArchiveAttendance	Yes	Occurs every 1 day(s), at 2:00:00 AM.

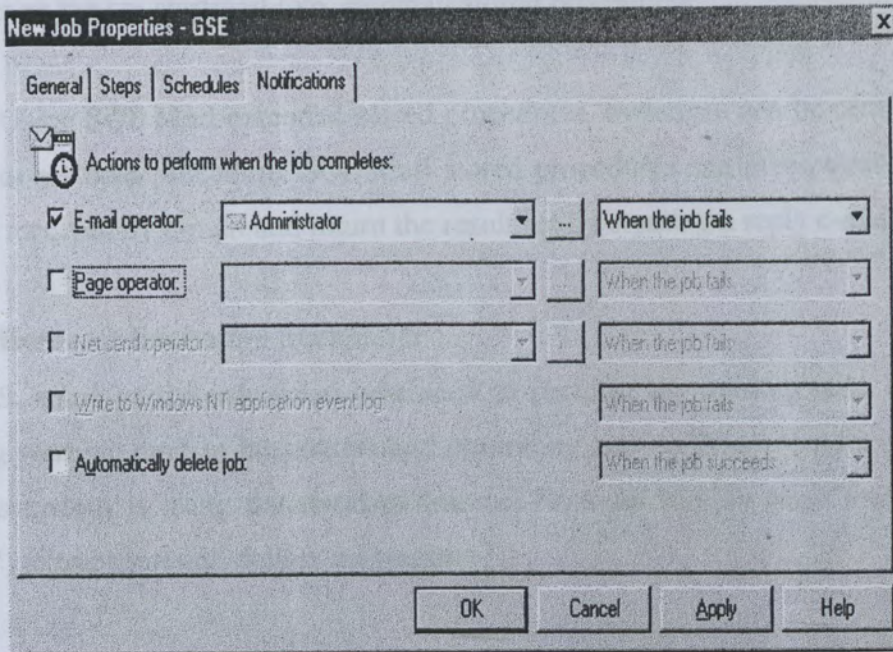
New Schedule... New Alert... Edit... Delete

OK Cancel Apply Help

**Figure 5-8 Completion of Job Schedule**



17. Lastly, under **Notifications** tab, select the actions to be performed when the job completes. For ArchiveAttendance, administrator will be notified by e-mail upon the job failure.



**Figure 5-9 Job Notifications**

18. Click **Apply** followed by **OK**. The job has been configured and ready to be tested. Right click the job to start it. The SQL Server Agent must be started before the job can be run.

## SQL Mail

SQL Mail allows Microsoft SQL Server to send and receive e-mail by establishing a client connection with a mail server. Ms SQL Server processes mail for all of the mail stored procedures. SQL Mail establishes a simple MAPI connection with a mail host. It can connect with Microsoft Exchange Server, Microsoft Windows NT Mail, or a Post Office Protocol 3 (POP3) server.

The computer running SQL Server must be set up as e-mail client. This is done through the installation of Microsoft Outlook or Microsoft Messaging. After the installation, an e-mail profile must be specified in SQL Mail Service. The e-mail profile



for AMS is “**Administrator**”. The mail components of SQL Server can be started automatically when the SQL Server Agent service is started. Alternatively, the mail component can be started and stopped at will, using either SQL Server Enterprise Manager or the `xp_startmail / xp_stopmail` stored procedures.

Using SQL Mail extended stored procedures, messages can be sent from either a trigger or a stored procedure. SQL Mail stored procedures can manipulate data, process queries received by e-mail and return the result set by creating a reply e-mail.

#### 5.1.4 Microsoft Exchange Server 5.5

In AMS, some of the scheduled jobs need to perform the mailing task. For examples, sending warning mail to latecomers and reminding mail to those who forget to clock in. So, this system is using the standard Internet Protocol Simple Mail Transfer Protocol (SMTP) to transport and deliver messages.

In order to connect Microsoft Exchange Server 5.5 with SQL Mail, a simple MAPI connection needs to be established. Prior to this, Exchange Server needs to be configured. The first step is to install the Internet Mail Connector / Service. Internet Mail Service routes messages between Microsoft Exchange Server and Simple Mail Transfer Protocol (SMTP)-based systems, including Internet mail servers and Post Office Protocol version 3 (POP3) and Internet Message Access Protocol, Version 4rev1 (IMAP4rev1) clients.

After the installation of Internet Mail Service, set up a MAPI profile (mail profile). MAPI profile is a set of configuration options used by Microsoft Exchange and other MAPI compliance messaging application. When configured, mail profiles are specific to the Windows NT user domain account that is activated when a user logs on to Windows NT successfully. SQL Mail must have a mail profile created in the same user domain account or context that is used to start SQL Server. When a mail stored procedure is executed, SQL Mail looks for the defined mail profile in the domain account that triggered it. Below are **steps to configure mail profile (window NT)**:



1. Logon to the Windows NT server specifying the name and password used to start SQL Server services for that server.
2. Create and configure or copy an existing mail profile by clicking on the **Mail** icon in Control Panel. Select the appropriate mail service to interact with your mail host. For AMS, the chosen mail service is Ms Exchange Server and the name or IP address for this mail server is **202.185.107.20**. Then, choose a name for the profile that will help the recipient identify the messages. The **user profile name** for AMS is **Administrator**. Default port number is used for POP3 and SMTP protocols.
3. On the server running Microsoft SQL Server, start the mail client using the newly created mail profile.
4. Send a message addressed to the same profile name to ensure the mail client, mail profile and e-mail provider are working properly. If your e-mail message does not appear, you may need to force mail synchronization by going to the **Tools** menu, and then clicking **Deliver Now**.
5. Exit the mail client program.

## 5.2 Module Implementation

AMS has 2 main modules, which are the user and administrator modules. Two type of languages involved in developing this two modules, which are Visual Basic 6.0 and ASP 3.0. The following sections explain the details about the implementation of each module throughout the project.

### 5.2.1 User Module

User module is available for stand-alone system and web system. The stand-alone system is developed using Visual Basic while for the web system is ASP. In installing the stand-alone system, the system DSN named **amsdev** was registered through the ODBC in that stand-alone PC in order to map the database server.



### 5.2.1.1 Stand-alone System

The user module for stand-alone system consists of 4 forms, which are frmAttend, frmLogin, frmmain and frmview. All these forms are interacting with several sub-modules, which are authenticate logon module, daily attendance module, daily updating process module, view attendance module and system control module. All modules are integrated before the application is generated for installation.

#### Authenticate Logon

- This module is implemented to verify the login ID and password entered by the user. Password entered into the system would not be seen by anybody and is depicted by \*\*\*\*\*. This is to avoid anybody from hacking the system as well as to ensure the data integrity.
- To implement this type of feature, all the passwords in the database must be encrypted. As such, a function called **EcPass (password as Variant)** is created. The password entered by a user must be matched with the password in the database. This function encrypts the password in the numeric format. The codes are shown as below:

#### Public Function EcPass(password As Variant)

```
Dim i, iLen, AscVal, TempPass, TempChar, ReversedStr
```

```
HexStr = ""
```

```
RetVal = 0
```

```
TempPass = Trim(password)
```

```
ReversedStr = StrReverse(TempPass)
```

```
iLen = Len(ReversedStr)
```

```
For i = 1 To iLen Step 1
```

```
    TempChar = Mid(ReversedStr, i, 1)
```

```
    AscVal = AscVal + Asc(TempChar) + 10041970
```

```
Next
```

```
EcPass = AscVal
```

#### End Function



- Beside that, when a user login to the system, this module will check the position code of the user. Only the non-academic staffs with status 1 can use the stand-alone system. This is implemented through the codes from **sub cmdlogin\_click ()** as shown below:

#### **Private Sub cmdlogin\_Click()**

Dim TempPass As String

Dim LoginID As String

On Error GoTo ErrorHandler

If validation Then

TempPass = EcPass(txtpass.Text)

newset

strqry = "Select A.DepartmentCode, A.fullname, A.applicantID, B.status " & \_  
 "from applicantinfo A, positioncode B where " & \_  
 "A.LoginName = " & Trim(txtlogin.Text) & " " & \_  
 "AND A.password = " & Trim(TempPass) & " " & \_  
 "AND A.LoginName = " & Trim(txtlogin) & " " & \_  
 "And A.positioncode = B.positiontypecode "

rsttemp.Open strqry, cnn, , , adCmdText

If Not rsttemp.EOF Then

userid = rsttemp!ApplicantID

DepCode = rsttemp!DepartmentCode

username = Trim(rsttemp!fullname)

'IIf (condition, truepart, falsepart) - return one or two parts

ustatus = IIf((rsttemp!Status = True), 1, 0)

rsttemp.Close

If ustatus = 0 Then

**MsgBox "Academic Staff are not allowed to login here. "** & \_

"Please use your own PC.", vbExclamation, TITLE

cmdreset\_Click

Else

frmmain.Show

End If

Else

**MsgBox "Invalid LoginID or Password. Please try again.", vbExclamation,**  
**TITLE**

txtlogin.Text = ""

txtpass.Text = ""

txtlogin.SetFocus

End If

End If



ErrorHandler:

```

If Err.Number <> 0 Then
    If Err.Number <> 364 Then
        MsgBox Err.Number & " " & Err.Description, vbCritical, TITLE
    End If
End If

```

End Sub

- If the particular day is holiday or the user is taking leave on that day, the user cannot login to the system.
- The rest of the functions for this module are:
  - Sub validation ()
  - Sub detectstatus ()
  - Sub detectday ()
  - Sub detectholiday ()

### Daily Attendance

- This module is subsequently implemented after the Authenticate logon. It is used to verify and update the daily attendance process like clock in, clock out, overtime clock in and overtime clock out.
- The clock in and clock out time of user will be based on the system time of local PC. The record is updated using the ADO 2.0 reference library.
- It is implemented through a sub function called **Sub retrieve (tag As Integer)**. This function will do a checking before the user's clock in/out time is updated to avoid the user from clocking in/out twice a day. Partial of that function is shown as below:

#### Sub retrieve(tag As Integer)

```
Dim strqry As String
```

```
On Error GoTo ErrorHandler
```

```
Select Case tag
```

```
Case 1 'clock in checking / updating
newset
```

```
strqry = "Select * from Attendance where ApplicantID = " & userid & _
" AND Date = " & Format(Now, "mm/dd/yyyy") & ""
```



```

rsttemp.Open strqry, cnn, , , adCmdText
With rsttemp
If Not .EOF Then
    If Not IsNull(!clockin) Then
        MsgBox "You have already clock in", vbExclamation, TITLE
        .Close
        Exit Sub
    Else
        !clockin = Format(ontime, "hh:mm")
        !Status = typeattend ' Present or late
        !IPAdd = "Station Login"
        If reason <> "" Then
            !remark = reason
        End If
        .Update
        .Close
        MsgBox "Your clock in time is " & ontime & ".", vbInformation,
            TITLE
        Unload Me
        frmAttend.Show
        frmAttend.txtlogin.Text = ""
        frmAttend.txtpass.Text = ""
        frmAttend.txtlogin.SetFocus
    Else
        MsgBox "Record not found. Please contact the administrator.",
            vbExclamation
        .....
        .....
        .Close
    End If
End With

Case 2 'clock out checking / updating
.....
Case 3 'overtime clock in checking / updating
.....
.....
End Select
.....
.....
End Sub

```



### Daily Updating Process

- This module will only be called when there is a power failure or the database server is down whereby the attendance records are not created in database.
- When the first person comes in to login to the system, it will create the attendance records and detecting those who are taking leave. Then, it will update the records to attendance table.
- The following are the functions for this module:
  - Sub DetectLeave ()
  - Sub DetectHalfLeave ()
  - Sub createRecord ()

### View Attendance

- This module is implemented to view the daily attendance record.
- It is using the List View control form from Visual Basic to display the attendance list.
- The following are the functions for this module:
  - Sub DisplayAttend ()
  - Sub DisplayOT ()
  - Sub DisplayRow ()

### System Control

- This module is implemented to prevent the system from being hacked by users and also to avoid the stand-alone machine being shut down by anyone.
- API calling function is used to perform some windows tasks for this module. For an example, to disappear the start button of the stand-alone machine, it can be done by set the HideStartButton function to true, **HideStartButton(True)** as shown below:



**Public Sub HideStartButton(ByVal visible As Boolean)**

```

Dim hWnd As Long 'The application windows's handle
Dim ClassName As String
Dim RetVal As Long

```

```

'Shell_TrayWnd - classname of window 95 task bar
hWndStart = FindWindow("Shell_TrayWnd", vbNullString)

```

```

'Get first child window
hWnd = GetWindow(hWndStart, GW_CHILD)

```

```

Do Until hWnd = 0
    ClassName = Space$(255)
    RetVal = GetClassName(hWnd, ClassName, 255)
    ClassName = Left(ClassName, RetVal)
    If (ClassName Like "Button") Then
        If visible Then
            RetVal = ShowWindow(hWnd, SW_HIDE)
        Else
            RetVal = ShowWindow(hWnd, SW_SHOWNORMAL)
        End If
    End If
    'Get next child window
    hWnd = GetWindow(hWnd, GW_HWNDNEXT)
Loop

```

**End Sub**

- However, this module does provide the function to enable the start button of the machine. Only the administrator is empowered to do that. This feature is implemented in the system by setting the DisableCtrlAltDelete function to True.

**Public Sub DisableCtrlAltDelete(bDisabled As Boolean)**

```

Dim X As Long
X = SystemParametersInfo(97, bDisabled, CStr(1), 0)

```

**End Sub**



### 5.2.1.2 Web-based System

The user module for web-based system also consists of several sub-modules, which are **authenticate logon module**, **daily attendance module**, **view attendance module**, **calendar module**, **report module**, **backward overtime module** and **overtime conversion module**. All modules are developed using the VB Script and HTML language. The entire module is implemented individually before there are integrated together.

The user module implemented for the web-based system is similar to the stand-alone system. The only difference is it is integrated with leave management system. When a user logon into the system, there are options provided either to access to attendance system, leave system or some other functions. Beside that, it contains a check status module that will check whether the particular day is holiday and whether the user has already clock in or clock out and display the control that the user haven't perform.

In web-based system, the user module uses **activeX DLL** and server **object CDONTS** for some of the functionalities. For an example to display the current datetime in the screen - Sunday 20 January, 2001; Time - 20:15. ASP does not provide this type of datetime format. So, functions for converting datetime format can be written in VB class module and make it to dll file. In AMS, this dll file is called '**convertse.dll**'. It must be registered to system server 32 before it can be used. Only after that, the **convert.format** object in ASP can recognise and use the corresponding datetime format that has been written in VB class module. There are a lot of formats which can be converted in ASP like "dd/mm/yy", "hh:mm", "mmm/yy", etc. Below is an example function written in VB class module to be made into dll file.

**Public Function datetoday(Intime As Variant) As Variant**

**datetoday = Format(Intime, "dd/mm/yyyy")**

**End Function**



After registering the dll file, Convert.format object in ASP can recognize the datetime format. Below are the codes for using the datetoday format (dd/mm/yyyy) in ASP.

If rsttemp.fields("Date") = 1-12-00, after using the Convert.Format object, mydate will become = 01/12/2000 in ASP.

```
<%  
    dim mydate  
    set objdate = Server.CreateObject("Convert.Format")  
    mydate = objdate.datetoday (rsttemp.Fields ("Date"))  
%>
```

Other than that, AMS needs to send mail to users in some of the user's sub modules like **backward overtime** and **overtime conversion**. Mails would be send if their half-day leave conversion applications have been approved. In this case, an object called **new mail object** must be created. The following code is an example using this object to send mail:

```
<%  
    set objMail = server.CreateObject("CDONTS.newmail")  
    objMail.From=emailfrom  
    objMail.To= emailto  
    objMail.BodyFormat=0  
    objMail.MailFormat=0  
    objMail.Body=msg  
    objMail.Subject = "Approved Half Day Leave Conversion"  
    objMail.send  
    SET objMail= NOTHING  
%>
```

In order to provide users with an online calendar that can acts as their personal diary, **calendar module** will be implemented in web-based system. To implement this module, calendar will be drawn using HTML codes as well as vbscript. User can add details into the calendar, which will be updated to database. In return, they can view their calendar at any time and anywhere they like. Below are the codes of drawing the calendar in ASP:



```

<% REM -----BEGINNING OF DRAW CALENDAR SECTION----- %>
<% REM This section executes the event query and draws a matching calendar. %>
<%
    Dim iDay, iWeek, sFontColor, dictDte(31,2), intCount
    strSql = "SELECT * FROM Calendar WHERE month(dte)= " & month(dtCurViewMonth) & _
        " and year(dte)= " & year(dtCurViewMonth) & _
        " and ApplicantID = " & session("AppID") & " order by dte"

    set rs = my_conn.Execute (StrSql)
    intCount= 0

    ' populate array with days of month
    do until rs.EOF
        if Day(rs("dte")) = intCount + 1 then
            dictDte(intCount, 1) = rs("text_field")
            rs.MoveNext
        Else
            dictDte(intCount, 1) = " "
        End If
        dictDte(intCount, 2) = intCount + 1
        intCount = intCount + 1
    loop
%>
<html>
<body bgcolor="ANTIQUEWHITE">
<form NAME="fmNextPrev" ACTION="diary.asp" METHOD="POST">
<center><font face="verdana" size="2" color="purple">
    You may insert details into this calendar by clicking on the date.</font></center>
<table align = center CELLPADDING="2" CELLSPACING="0" WIDTH="80%"
    BORDER="2" BGCOLOR=b0c4de BORDERCOLORDARK="#003399"
    BORDERCOLORLIGHT="#FFFFFF">
<tr VALIGN="MIDDLE" ALIGN="CENTER"><td COLSPAN="7">
    <table CELLPADDING="0" CELLSPACING="0" WIDTH="100%" BORDER="0">
<tr VALIGN="MIDDLE" ALIGN="CENTER">
<td WIDTH="30%" ALIGN="RIGHT">
<input TYPE="IMAGE" NAME="subPrev" SRC="Left.gif" BORDER="0" WIDTH="18"
    HEIGHT="20" HSPACE="0" VSPACE="0">
</td>

.....
.....

<%REM -----END OF DRAW CALENDAR SECTION-----
    my_conn.Close
    set my_conn = nothing
%>

</html>

```



## 5.2.2 Administrator Module

Initially, the administrator module is developed in both the stand-alone and web-based system. In the stand-alone system, administrator can configure the daily attendance and generate report using the feature of crystal report in Visual Basic, which this cannot be done in ASP. However, the stand-alone system needs to be installed into the client machine and it needs another extra license to run the system. Further more, it requires another Crystal report license to perform the report printing. This will increase the cost needed to perform daily operation. As a conclusion, the administrator module now only consists of one system, which is the web-based system.

### 5.2.2.1 Web-based System

The administrator modules of web-based system consists of several sub-modules, which are attendance setting module, working hours setup module, report module and add hot site module. All modules are developed using the VB Script and HTML language. The entire module is implemented individually before there are integrated together.

#### Attendance Setting

- This module will be implemented to allow administrator to view, add, delete or update the attendance records of any staffs in the faculty. This can be done by mapping the module with the data in Attendance and Yearly Attendance table in database. Thus, many functions are written for the implementation of this module. Update function is chosen here as an example to show how administrator can modify one's attendance record at a particular date.
- Before updating the data, there must be a connection between the client-side application and database server. To connect to the database server, a connection string must be provided with the **DSN name**, **UID** and **password** of the database server. In AMS, the connection string is provided in Application ("AMSDB"). After that, in the asp file, create the connection using two statements below:

```
Set cnn = Server.CreateObject("ADODB.Connection")  
Cnn.Open Application ("AMSDB")
```



- The modified record of administrator would then be updated to database through the combination of the above two statements and the codes below:

```
<!-- #include file = "adovbs.inc" -->

<%
set cnn = Server.CreateObject ("ADODB.Connection")
cnn.Open Application("AMSDB")
set rsttemp = Server.CreateObject ("ADODB.Recordset")

strqry = "Select * from Attendance where FullName = " & session("username") & _
        "" AND Date = " & objdate.datetomonth (date) & ""
rsttemp.Open strqry,cnn,adOpenDynamic, adLockOptimistic,adcmdText

if not rsttemp.EOF then
    rsttemp("clockin")= objdate.datetotext(session("temptime"))
    rsttemp("Positiontypename")= session("usergroup")
    rsttemp("Status") = session("status")
    rsttemp("uStatus") = session("userstatus")
    rsttemp("IpAdd") = session("IpAdd")
    If session("status") = Application("StaLate") Then
        rsttemp("remark") = trim(session("reason"))
    end if
    rsttemp.Update
    rsttemp.Close
end if
%>
```

### Working Hours Setup

- Administrator can view, add, delete or update the working hours setup in this module. The method in implementing this module is same with the Attendance Setting module. The only different is the table being used. Working hours setup module will map the TstTimeSetUp table to its client-side application.

### Report

- Two main functions provided under this module are full report viewing and summary report viewing. The report is generated according to the selection of administrator, which is based on the criteria such as date, name and type. The key in implementing this module is the **manipulation of query strings**. There are many query strings that



have been written in this module. Below is an example to query the summary report based on the selected date:

```
strqry1 = "Select count(*) as subtotal, A.Status, B.Fullname "& _
         "from Attendance A, Applicantinfo B, DeptCode C " & _
         "where A.applicantid = B.ApplicantID "& _
         "and B.DepartmentCode = C.DepartmentCode " & _
         "and A.Date >= '" & tempstart & "' and A.Date <= '" & tempended & "' "
```

```
strqry2 = "Select count(*) as subtotal, A.Status, B.Fullname "& _
         "from [Yearly Attendance] A, Applicantinfo B, DeptCode C " & _
         "where A.applicantid = B.ApplicantID "& _
         "and B.DepartmentCode = C.DepartmentCode " & _
         "and A.Date >= '" & tempstart & "' and A.Date <= '" & tempended & "' "
```

```
strqry1 = strqry1 & " Group by A.Status, B.Fullname " & _
         "Union all " & _
         strqry2 & " Group by A.Status, B.Fullname " & _
         "Order by Fullname "
```

## Add Hot Link

- In implementing the add hot link module, the existing of the hot link address in database as well as the connection string must be taken into consideration. If there is a duplicate hot link address in database, the system must be able to detect it and pop up an error message. The following are the partial codes in this module:

```
<%
```

```
Dim tag, appweb, applocation, apptype
```

```
tag = Request.Form("txttag")
```

```
if tag = "Add" then
```

```
    appweb = Request.Form ("txtweb")
```

```
    applocation = Request.Form ("txtlocation")
```

```
    apptype = Request.Form ("cbotype")
```

```
    if updatedata then
```

```
        session("updateapply") = "The record has been updated successfully"
```

```
        Response.Redirect "addlink.asp"
```

```
    else
```

```
        session("updateapply") = "Error! Duplicate record has been found."
```

```
        Response.Redirect "addlink.asp"
```

```
    end if
```

```
else
```

```
    session("updateapply") = "Database Error! Please try again"
```

```
    Response.Redirect "addlink.asp"
```

```
end if
```



**Function updatedata**

on error resume next

```

updatedata = false
rsttemp.Open "HotLink", cnn, adOpenDynamic, adLockOptimistic, adcmdtable
rsttemp.AddNew
rsttemp("UrlAdd")= applocation
rsttemp("Name")= appweb
rsttemp("IntType")= apptype
rsttemp.Update
if err.number <> 0 then
    rsttemp.Close
    exit function
end if
rsttemp.Close
updatedata = true

end function
%>

```

**5.2.3 Integrated Modules**

Besides the user module and the administrator module, some of the AMS modules have been modified to be integrated with Leave Management System. The module is developed together in order to reduce any duplicate function as well as prevent any confused between the two systems. The integrate modules are change password module, calendar module, attendance module and leave module.

With the integration, users can now access to the both system by just logging in once not twice as before the integration. This is implemented through the codes below, whereby users can select either to proceed with the clock in, attendance, leave or other transactions.

```

<% if session("holiday") = true then %>
<% else %>
    <%If sta = 0 and session("userstatus") <> "" then %>
        <div class="title" id="title1" style="top: 85px">
            <a href="in.asp?AttendType=Clock+In" target="mright">
                <font Color="darkred">Clock In</font></a>
        </div>
    <% tag= true %>

```



```

<%elseif sta = 1 and session("userstatus")= "True" then %>
.....
<%end if %>
<% if sta = 2 or sta = 3 or ( sta = 1 and session("userstatus")= "False" ) or_
session("userstatus")= "" then%>
    <div class="title" id="title1" style="top: 85px">
        <A href="AttendLeft.asp" target=_top>
        <font Color="darkred">Attendance</font></A>
    </div>
<%else%>
.....
<%end if%>
<% end if %>

<% if session("holiday") = true then %>
    <div class="title" id="title2" style="top: 85px">
        <A href="check.asp" target="datawindow">
        <font Color="darkred">Leave</font></A>
    </div>
<% else %>
.....
<%end if%>
.....
</body>
<%else
    response.redirect "frontframe.html"
end if%>

```

### 5.3 Conclusion

System implementation phase is critical in ensuring the success of the project. All the development platforms and modules are implemented successfully in this phase. The next chapter will discuss about the system testing.



## CHAPTER 6 : TESTING

Testing is done parallel with the system development and system implementation. It is important to ensure that the final product is reliable and performs as what it should be. So, before the AMS is released, testing has been done in several stages to ensure that the system is developed according to its specification and every function implemented in the application works properly.

Testing in this phases includes unit testing, module testing, interface testing, integration testing and finally system testing.

### 6.1 Unit Testing

Unit testing technique is done independently when a new module is developed. It deals directly with the structure of codes within a module or segment to assure its accuracy and correctness. It is important to fix the bugs and faults starting from each unit before they could result in the whole system failure. Three kinds of testing strategies have been carried out for the unit testing. They are:

- Code Reviewing and Testing
- Input Testing
- Data Flow Testing

#### 6.1.1 Code Reviewing and Testing

Under this strategy, codes are reviewed and tested line by line in order to make sure that any uncovered semantic or syntax errors could be revealed. The correctness of coding is identified by comparing it to the original design of the program flow.

All the codes both in stand-alone system or web-based system can be debugged and traced using the debugger. With the facility of debugger, the value of any variables can be seen just by highlighted the variable and press shift+F9. However, there are other ways in testing the codes. For ASP, the "Response.Write" command is inserted into the



code to exam the value of the variable. For an example, the code below shows that 6 was selected as date from the combo box:

```
<%
    count = Request.form ("cbodate")

    if count < 8 then
        call detecting
    end if
%>
```

This code can be debugged and tested as below:

```
<%
    count = Request.form ("cbodate")

    Response.write count      ' this will print out the value of valuable count

    If count < 8 then
        Response.write "Run" ' this statement will be printed out if the if condition if correct.
        Call detecting
    End if
%>
```

By using the "Response.write" command, a value can be printed on the browser. Thus, it is used as the 'watch' value of the variable. In this example, the value 6 was printed out. Since the value is smaller than 8, it will call the function detecting and printed out the word "Run". The word "Run" is used to check and test the if statement. If the value 6 and the word "Run" is not printed out, that means there is something wrong with the code. However, if the code is simpler and written with full of confidence, using this technique to test the code is just a redundant work

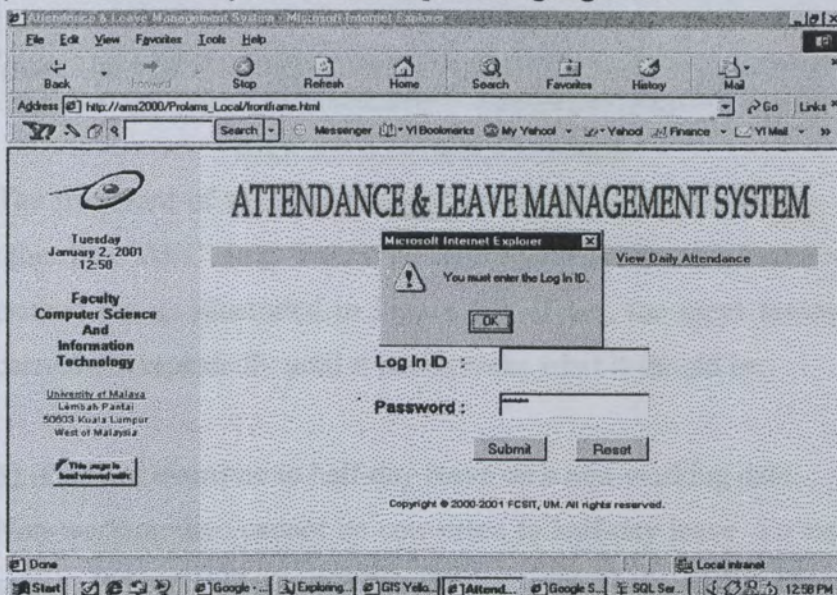


### 6.1.2 Input Testing

This approach is used when a set of input is given and its output is observed. In this strategy, different kinds of input are provided to test the compound statement. Every stated condition must be able to be executed and any unnecessary codes are eliminated. It is important to identify the variance between the prototype and requirements.

For an example, a user deliberately does some mistake or key in the value that is out of acceptable range. The purpose is to see whether the system will react and prompt the user with appropriate error message. Below are the examples of input testing:

#### 1) Access to the system without providing login ID.

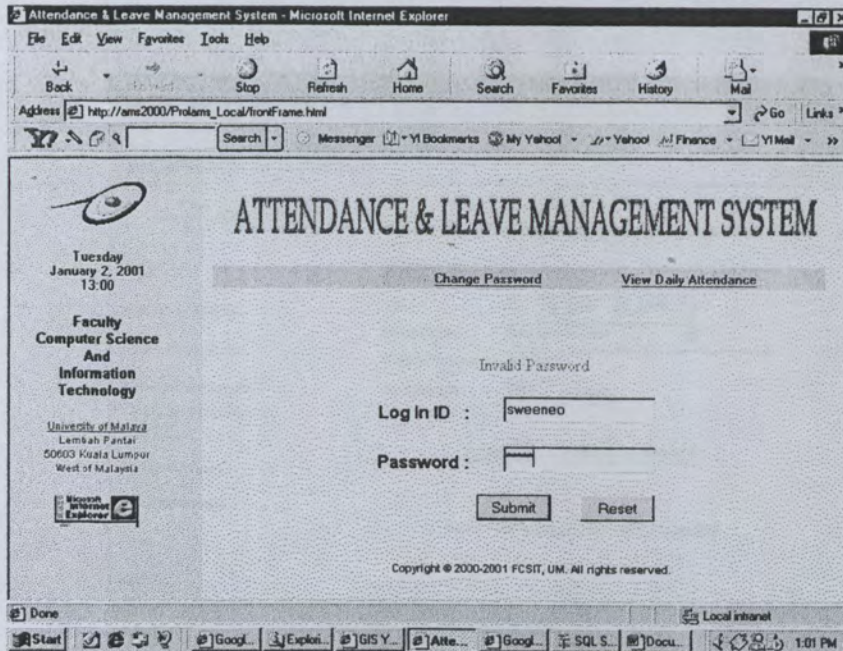


**Figure 6-1 Invalid Login ID**

As illustrated in Figure 6-1, an error message is popped up to prompt user to enter the login ID. This indicates that the code is right. If the user can access to the system without entering the login ID, this indicates that there are something wrong with the codes. Debugging and correctness need to be carried out.



## 2) Type in the wrong password

**Figure 6-2 Invalid Password**

The password of sweeneo in database is 'ABC', if she keys in the password other than 'ABC' (e.g: swee); an error message "Invalid Password" should be printed out as illustrated in Figure 6-2. If not, the input testing needs to be carried out repeatedly until it performs as what it should be.

## 3) Convert overtime to half-day leave on a non-working date.

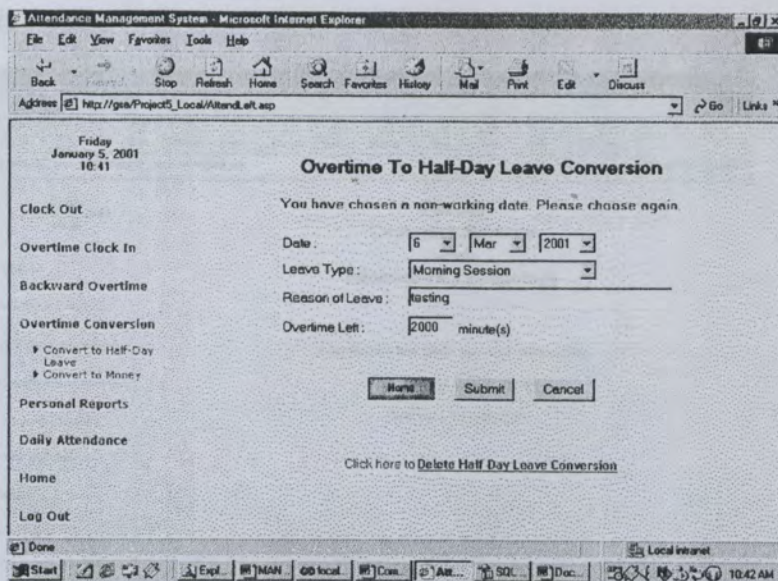
Non-working date cannot be converted to half-day leave. To test whether the system is functioning correctly, choose a non-working date from the database as illustrated in Table 6-1. Example is 3/6/01 which is Hari Raya Haji.

SQL Server Enterprise Manager - [2:Data in Table 'NonWorkingDate']			
Console Window Help			
SQL			
NonWorkingDate	Description	HolidayType	
3/6/01	Hari Raya Haji	PUB	←
1/20/01	3rd Saturday of January	ORG	
6/2/01	Birthday of DYMM SPB Yang Di-Perl	PUB	
8/31/01	National Day	PUB	
12/16/01	Hari Raya Puasa	PUB	
12/25/01	Christmas Day	PUB	

**Table 6-1 Non-working date**



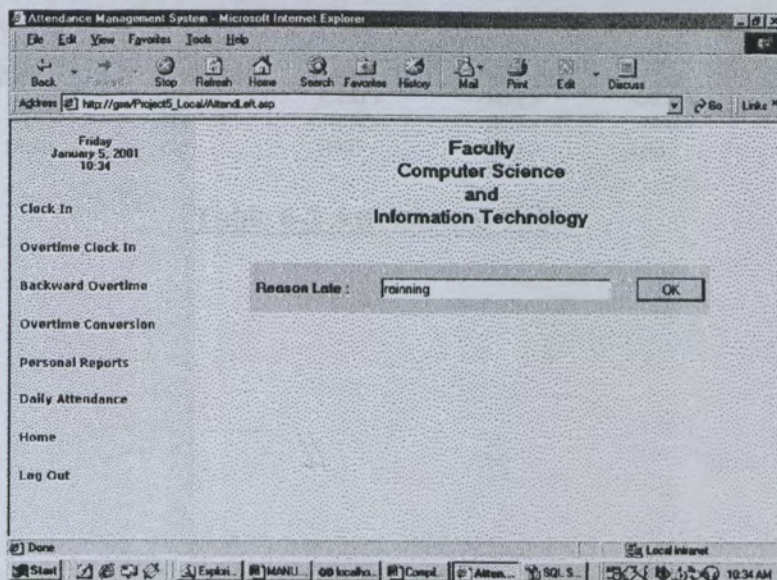
If the output is illustrated in Figure 6-3, the testing is considered to be a success.



**Figure 6-3 Invalid Date Selection**

### 6.1.3 Data Flow Testing

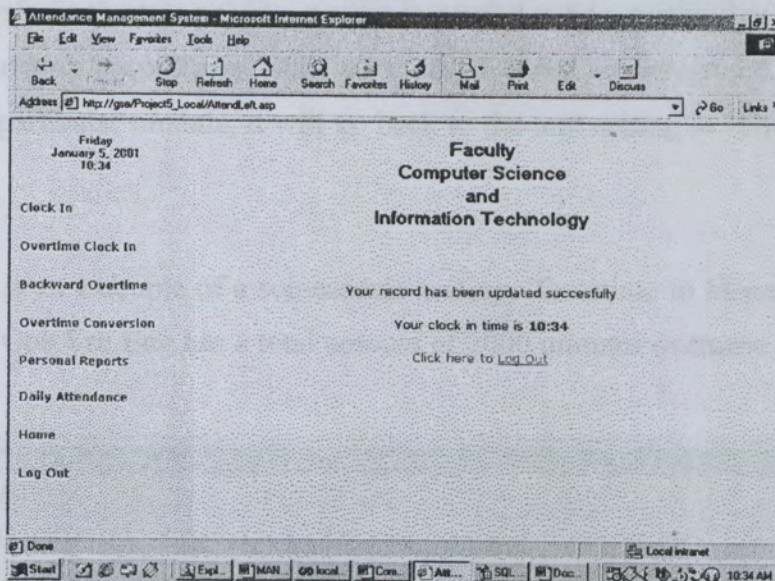
Under this strategy, the unit testing is dealing with the data flow. The codes are tested whether they are executed in the correct sequence and order. For an example, a user with the applicant ID 213, came late for work and clocked in at 10:34 am; giving raining as reason as illustrated in Figure 6-4.



**Figure 6-4 Enter Reason for Late**



Upon the click of button OK, the system should display the clock in time confirmation. If the system does not display the page as shown in Figure 6-5, the data flow of the code is wrong.



**Figure 6-5 Clock In Time Confirmation**

Beside that, the “Attendance” table in database must be examined to check whether the data is updated successfully. Since the data is found in the third row of the table as illustrated in Table 6-2, the testing is completed and successful.

TransID	Date	ApplicantID	ClockIn	ClockOut	Status	Remark	Log
23013	1/5/01	203	<NULL>	<NULL>	<NULL>	<NULL>	<A
23014	1/5/01	211	<NULL>	<NULL>	<NULL>	<NULL>	<A
23015	1/5/01	213	10:34	<NULL>	Late	raining	<A

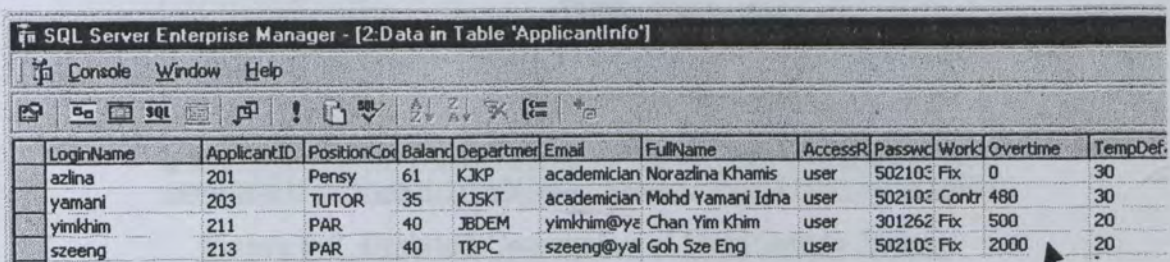
**Table 6-2 Attendance Table**



## 6.2 Module Testing

Module testing is continued from the unit testing. Every sub-module in both the user and administrator module is tested carefully. An administrator and some users are created in database to perform the testing. The testing is carried out to ensure that the code under the module function accordingly when all units of codes are integrated. If any error is discovered in particular module, it will go back to the unit testing to identify and fix the errors.

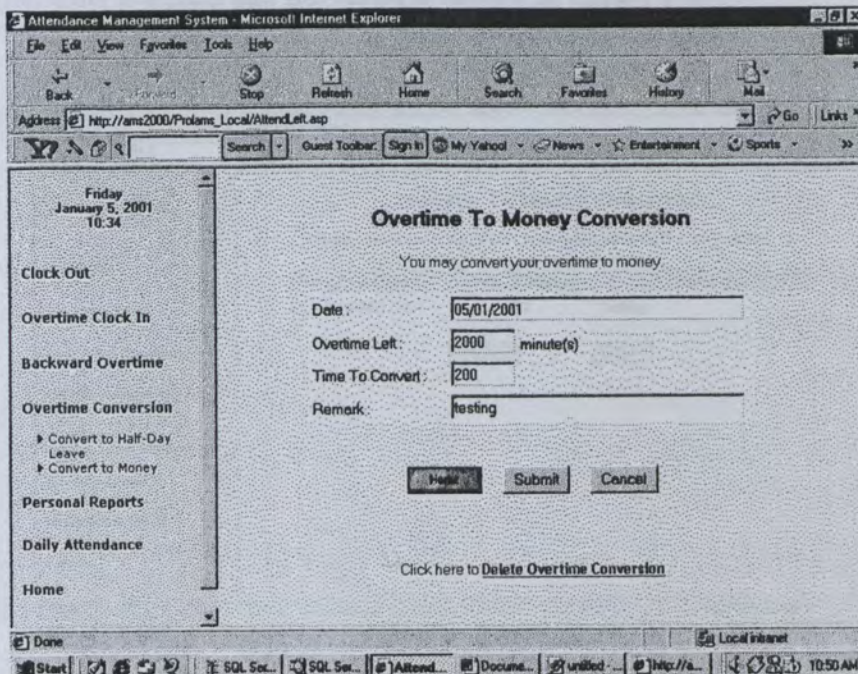
Below is an example of a sub-module testing, 'Overtime to Money Conversion'. A user named Goh Size Eng has a total amount of 2000 minutes overtime as illustrated in Table 6-3.



LoginName	ApplicantID	PositionCod	Balance	Department	Email	FullName	AccessR	Passwd	Work	Overtime	TempDef
azlina	201	Pensy	61	KJKP	academician Norazlina Khamis	user	502103	Fix	0	30	
yamani	203	TUTOR	35	KJST	academician Mohd Yamani Idna	user	502103	Contr	480	30	
yimkhim	211	PAR	40	JBDEM	yimkhim@ye Chan Yim Khim	user	301262	Fix	500	20	
szeeng	213	PAR	40	TKPC	szeeng@yal Goh Size Eng	user	502103	Fix	2000	20	

**Table 6-3 Before Overtime to Money Conversion**

A testing is carried out to convert 200 minutes of overtime to money as shown in Figure 6-6.



**Overtime To Money Conversion**

You may convert your overtime to money

Date: 05/01/2001

Overtime Left: 2000 minute(s)

Time To Convert: 200

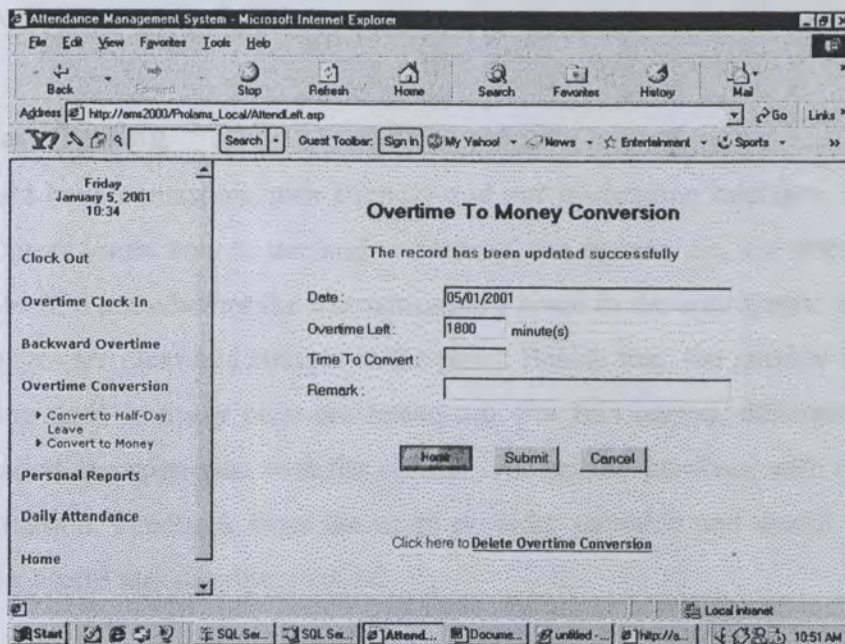
Remark: testing

Click here to [Delete Overtime Conversion](#)

**Figure 6-6 Conversion of 200 Minutes Overtime**

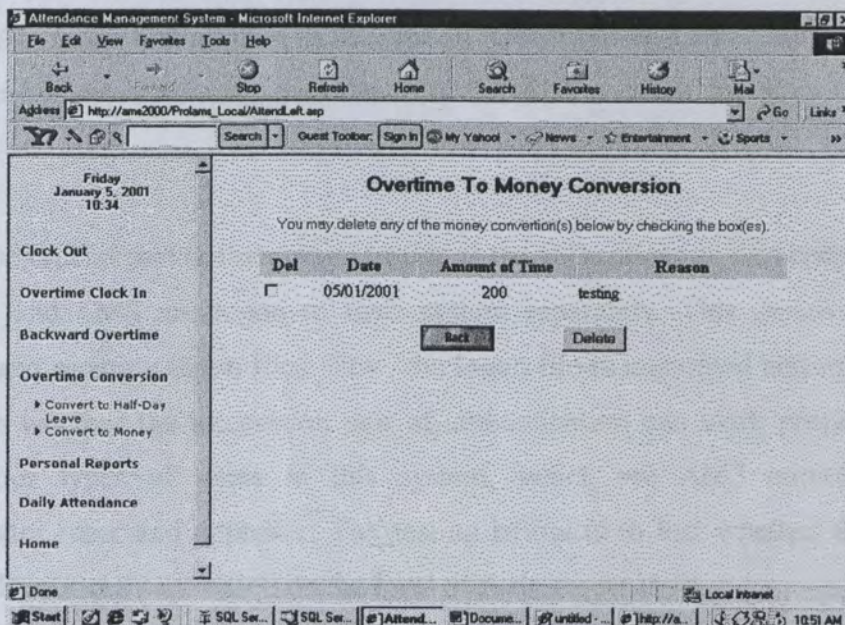


After the submit button is clicked, the total of overtime will be deducted. The overtime left now should equal to 1800 minutes. If the result of testing is as shown in Figure 6-7, it is correct.



**Figure 6-7 Updated Conversion of Overtime to Money**

Before the conversion is approved, the user can delete the conversion by clicking the link 'Delete Overtime Conversion' at the bottom of the page. The amount of conversion must be the same. This is tested and the result is shown in Figure 6-8.



**Figure 6-8 Unapproved Overtime to Money Conversion**



The approval of the overtime to money conversion is under the integration testing. This will be shown in section 6.4 whereby the conversion will be approved by the head department in leave management system. The final testing result in database (the actual amount of overtime left) is shown in Table 6-4 of section 6.4.

### 6.3 Interface Testing

AMS should have consistent, user friendly and not misleading interface. It is crucial to ensure all users know how to use and understand the system. So, the interface testing is important to find out whether the instructions are given in the appropriate manner and the error messages are clear and straight to the point. Beside that, the graphic designs, colour and wording used in every page are tested too. For this reason, different categories of user are called to experience with the system. The testing involved with random data in random situation. Feedback from the users provides valuable and useful information in producing a useful and reliable product.

### 6.4 Integration Testing

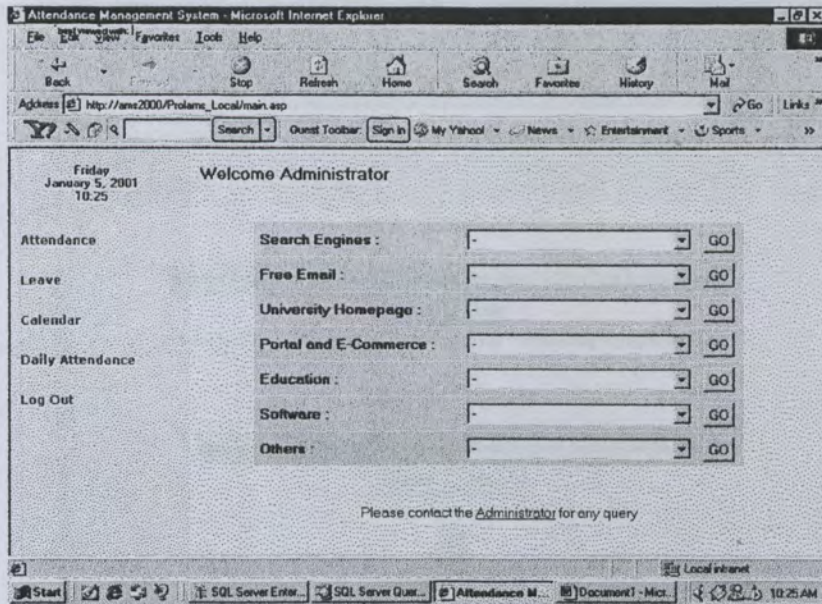
In this stage, all the modules in AMS, which have been tested separately, will be integrated with leave modules from LMS to form a complete system. This testing is to ensure that all modules that integrated together are able to interact and work seamlessly together. Approach that applied in integration testing is Top-Down Integration approach, where the highest-level main module is tested first and sub modules are then added gradually.

The crucial part in integration testing is the authentication part because previously the AMS and LMS are logon to their system separately. This means there are two different and separate logon files. Now, the logon file is integrated and testing is carried out to see whether the integration part of authentication can work properly. There are mainly four types of users in this system, which are AMS administrator, LMS administrator, user and approver. The testing below is to test whether the system will direct to the correct page based on the level of their access right.



## 1) Log in of AMS Administrator

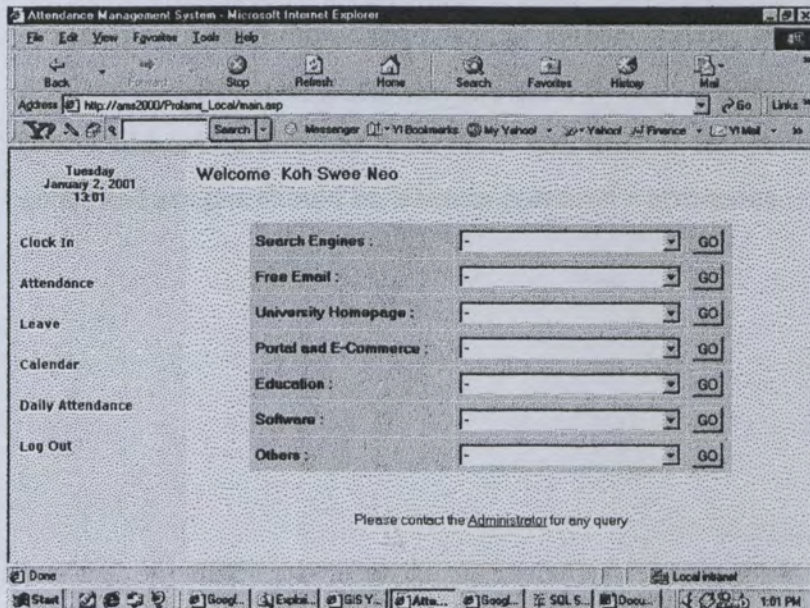
When the AMS administrator log in to the system, the testing is succeed if the administrator is directed to the page as illustrated in Figure 6-9.



**Figure 6-9 AMS Administrator's Welcome Page**

## 2) Log in of user

If the user is directed to the page as illustrated in Figure 6-10, the codes of integration part is free of error.

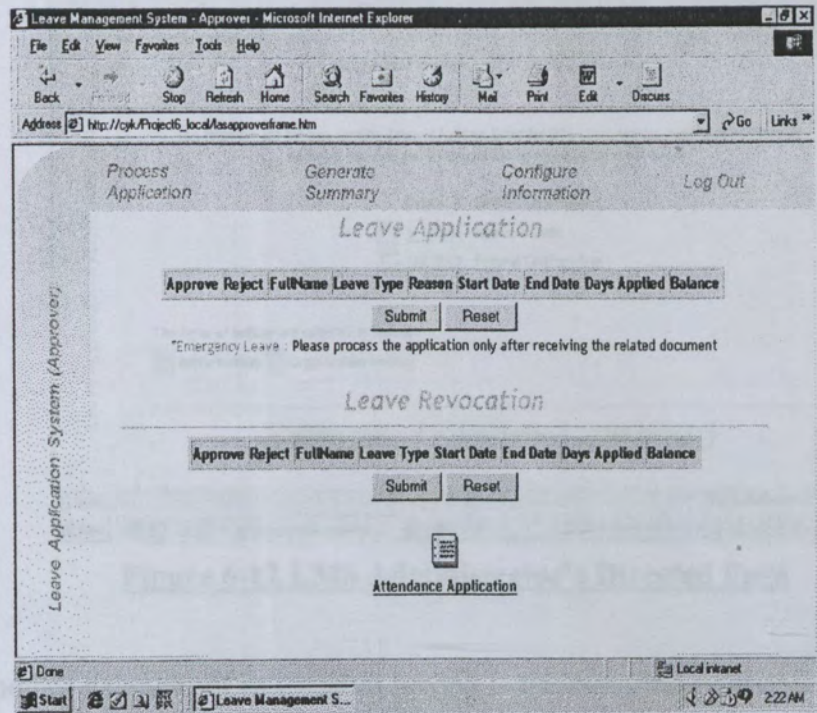


**Figure 6-10 User's Welcome Page**



### 3) Log in of Approver

When the approver log in to the system, if he or she will be directed to the approval page as illustrated in Figure 6-11, the code is correct.

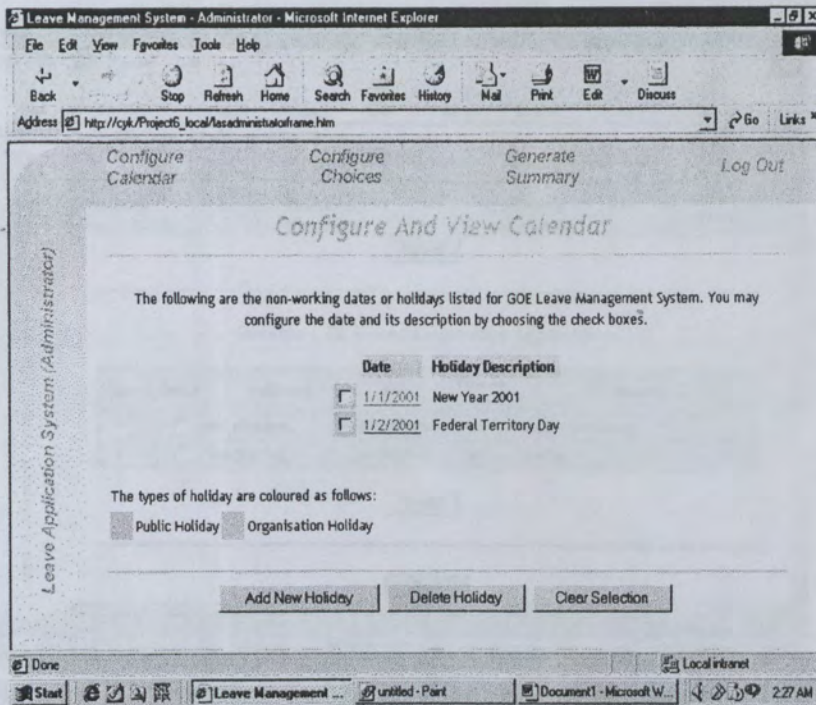


**Figure 6-11 Approver's Directed Page**

### 4) Log in of LMS Administrator

When the LMS Administrator log in to the system, he or she should be directed to the page as illustrated in Figure 6-12. It is a new window and he or she would not be directed to AMS. If not, there is something wrong with the codes and actions should be taken to correct the erroneous.



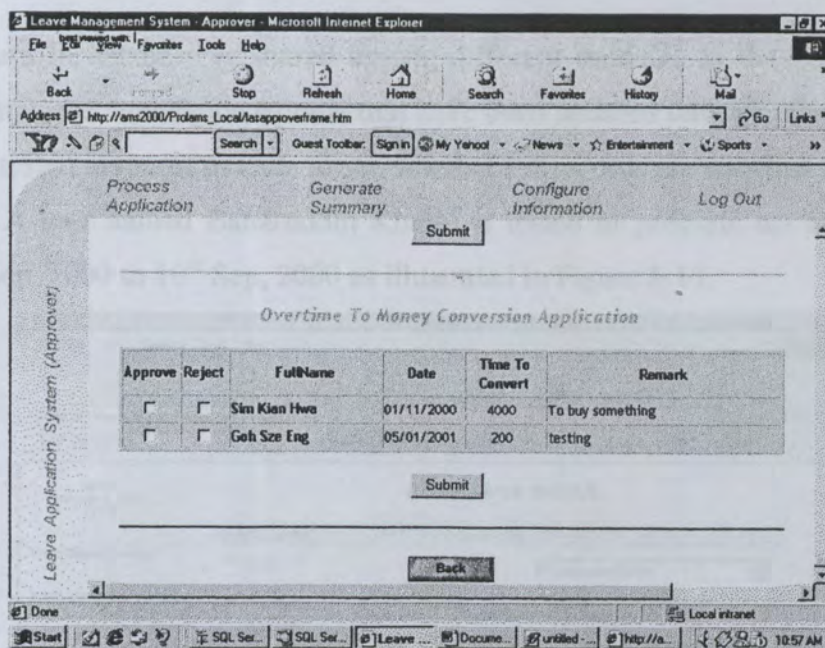


**Figure 6-12 LMS Administrator's Directed Page**

Another example of integration testing is the overtime conversion sub-module of AMS. As mentioned earlier in section 6.2, overtime to half-day leave or money conversion in AMS would be approved in LMS. All the calculations of overtime are tested to ensure that they are free of error. From the example in section 6.2, the conversion of 200 minutes overtime to money will be passed to the corresponding approver for approving. When the approver login into the system, he or she needs to click on the 'Approve Overtime and Half-Day Leave Application' link. A list of unapproved application of overtime to money conversion will be displayed on the screen. This is tested and the result is shown in Figure 6-13.

The total amount of overtime before the approval is 2000 (Table 6-3, section 6.2), after the conversion is approved, the overtime should leave 1800 minutes. This is proved to be correct as illustrated in Table 6-4 with a pointed arrow. Beside that, after the approval of the conversion, a letter will be sent to the applicant informing him or her that his or her overtime conversion has been approved.





**Figure 6-13 Overtime to Money Conversion Application List**

SQL Server Enterprise Manager - [2:Data in Table 'ApplicantInfo']

LoginName	ApplicantID	PositionC	Balance	Department	Email	FullName	AccessSt	Passwo	DateAc	WorkSt	Overtime	Temp
yamani	203	TUTOR	35	KJSKT	academician01	Mohd Yamani	user	502103	11/27/1	Contract	480	30
yimkhim	211	PAR	40	JBDEM	yimkhim@yaho	Chan Yim Khir	user	301262	1/1/01	Fix	500	20
szeeng	213	PAR	40	TKPC	szeeng@yaho	Goh Sze Eng	user	502103	1/1/01	Fix	1800	20

**Table 6-4 After Overtime to Money Conversion**

## 6.5 System Testing

Finally, the overall system testing is carried out to ensure the entire application can function properly and meet all the requirements. The testing also involved the scheduled job in the database server. Since the jobs are scheduled at different interval time and date, the system is tested according to the specified date and time by changing the server regional setting time. Beside that, some testers who are not the system developers are invited to experience with the system. They are assigned to different modules and do the testing concurrently. All the errors and faults are identified and jotted down to be corrected. Below are the examples of system testing that have been carried out:



## Test 1

All the data in database is shared among different modules in the system. Under the system test 1, the attendance records that have been updated through clock in/out module would be tested in report module to see whether the records are matched and correct as in database. A user named Baharuddin Khalid is tested to generate his attendance report from 7<sup>th</sup> Sep, 2000 to 16<sup>th</sup> Sep, 2000 as illustrated in Figure 6-14.

The screenshot shows a web browser window titled "Attendance Management System - Microsoft Internet Explorer". The address bar displays "http://ams2000/Proforma\_Local/AttendLeft.asp". The page content is titled "Attendance Report". On the left, there is a sidebar menu with options: "Attendance Setting", "Working Hours Setting", "Add Hot Sites", "Reports" (with sub-items: "Attendance Report", "Leave Report", "Overtime Report", "Activity Report"), "Daily Attendance", "Home", and "Log Out". The main content area has a "Date Selection" section with radio buttons for "Today", "Date Range", and "Monthly". The "Date Range" option is selected, showing "Start Date" as 7 Sep 2000 and "End Date" as 16 Sep 2000. To the right, a "Search By" section includes dropdown menus for "Name" (Baharuddin Khalid), "Department" (Bhg Teknikal (JND)), "Attendance Status" (All), "Logout Status" (All), and "Sort by" (Date). At the bottom of the main content area are three buttons: "View Report", "View Summary", and "Reset". The browser's status bar at the bottom shows the time as 11:23 AM.

**Figure 6-14 Generate Attendance Report**

The result is shown in Figure 6-15 after the **View Report** button is clicked. However, to prove that the records are correct as in database, a query is made through the SQL query analyzer as shown in Table 6-5. The testing is valid since the data in database is matched with the data in the attendance report.



Faculty Computer Science And Information Technology  
University Malaya

Attendance Report Start From 07/09/2000 To 16/09/2000

Name : Baharuddin Khalid  
Department : Bhg Teknikal (UNIX)

Date	Clock In	Clock Out	Status	Logout Status	Remark
07/09/2000	08:47	17:03	Late	Y	
08/09/2000	08:11	17:45	Present	Y	
09/09/2000	08:34	13:55	Present	Y	
11/09/2000	08:32	18:14	Present	Y	
12/09/2000	08:46	-	Late	N	Membuka makmal dan dewan kuliah dahulu.
13/09/2000	08:31	-	Present	N	
14/09/2000	08:28	17:33	Present	Y	
15/09/2000	08:16	17:11	Present	Y	

**Figure 6-15 Attendance Report**

SQL Server Enterprise Manager - [2:Data in Table 'Yearly Attendance']

Console Window Help

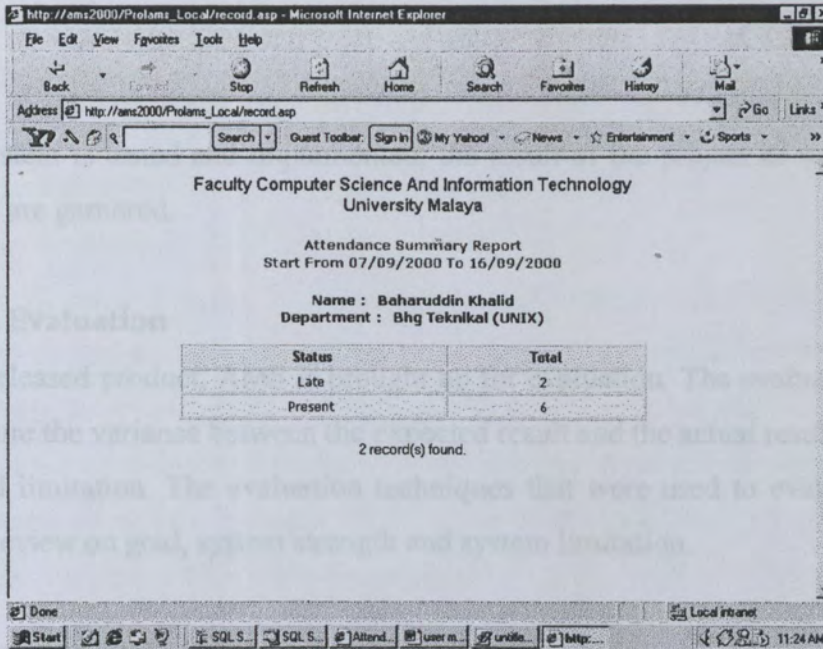
```
SELECT *
FROM [Yearly Attendance]
WHERE ApplicantID = 10 AND Date >= '9/7/00' AND
Date <= '9/16/00'
```

TransID	Date	ApplicantID	ClockIn	ClockOut	Status	Remark	Logout
14270	9/7/00	10	08:47	17:03	Late	-	Y
14347	9/8/00	10	08:11	17:45	Present	<NULL>	Y
14424	9/9/00	10	08:34	13:55	Present	<NULL>	Y
14501	9/11/00	10	08:32	18:14	Present	<NULL>	Y
14580	9/12/00	10	08:46	-	Late	Membuka makmal d N	N
14656	9/13/00	10	08:31	-	Present	<NULL>	N
14735	9/14/00	10	08:28	17:33	Present	<NULL>	Y
14814	9/15/00	10	08:16	17:11	Present	<NULL>	Y

**Table 6-5 Matching Data in Database**

Beside that, in this system testing, the view summary function is also tested to see whether the summary report is correct. From Table 6-5, Baharuddin Khalid is late for work for two days and is present for six days. The result of the testing shows that the summary report is correct as illustrated in Figure 6-16.





**Figure 6-16 Attendance Summary Report**

### Test 2

Stop the SQL server to create an unexpected condition like power failure or server down to see whether the system is still working in the next morning. In this case, all the daily scheduled jobs are failed to perform. So, a testing is done to replace the jobs from the web application. A tester will come and logon into the system to see whether the attendance records would be generated upon the logon of that tester. If records are generated, the testing is considered to be a success.

### Test 3

The system date is changed to the end of the year to test the Yearly Attendance scheduled job. This is done to test whether the yearly attendance table is archived to delete all the good records with attendance status of 'Present'. It is correct if the records are deleted.

## 6.6 Conclusion

After the testing phase, the system is running smoothly without critical failure. This is important to obtain user's confidence towards the system. The last chapter will be discussing about the project findings and conclusion.



## CHAPTER 7: PROJECT FINDINGS AND CONCLUSION

After the system is tested and implemented, the result of the project as well as findings and insights are garnered.

### 7.1 System Evaluation

As a final released product, AMS is brought up for evaluation. The evaluation is carried out to measure the variance between the expected result and the actual result as well as its strength and limitation. The evaluation techniques that were used to evaluate this final system are review on goal, system strength and system limitation.

#### 7.1.1 Review on Goal

Before the project is implemented, there are certain expectations on what should be achieved. At this final stage, a review of these expectations will give valuable insight.

##### 7.1.1.1 Proof of AMS Feasibility

AMS has fulfilled the expectation stated in the beginning stage of project. It is able to provide functions required by faculty as well as individual needs. Since the criteria like reliability, user-friendly, open system and wide accessibility have been met, AMS is feasible to be implemented and continuing its operation in faculty.

##### 7.1.1.2 Achievement of Objective

Most of the project objectives described in section 1.2 have been achieved at this final stage. This project has successfully created a system that supports the daily attendance process in the faculty coupled with its new features and functionalities.

##### 7.1.1.3 Foundation for Future Growth and Enhancement

The implementation of AMS created by this project has proven to be very capable in terms of scalability and flexibility for the future. New modules can be added as well as



the enhancement can be made to the current modules without affecting the main environment or other modules. The server architecture created now forms a foundation for the addition of more features into AMS.

### 7.1.2 System Strength

#### ➤ **Wide-accessibility**

AMS is developed under the web technology, which is using the client server approach that allows processing load to be shared between the client and the server. As a result, it is deployable not only at the stand-alone machine but also internet, users can access the system virtually at anytime and anywhere through out the world. What is needed in the client-side PC is just an Internet Explorer 4.0 or above which is preloaded with most of the operating system.

#### ➤ **Distributed Server Architecture**

The web server (IIS) and database server (SQL Server) of AMS are resided at different sever. This architecture allows added flexibility server implementation and lessened dependency on individual machines and future backup capabilities.

#### ➤ **Scalability**

AMS is not hardware dependency and is highly scalable. Additional hardware or software can be added easily without affecting the existing system. For an example, it is possible to add another machine dedicated as mail server in the future. The only change is the MAPI profile setting in the control panel and the DNS setting to recognize the mail server. However, this can be done if there is a trusted relationship for the machines in different domain.

#### ➤ **Reusability of Coding**

Since most of the functions in AMS are presented in the modular format, it makes them reusable and traceable. For example, the `updatedate()` function is not called by one particular file but will be called by many files like `in.asp`, `out.asp` and `otusage.asp`. Beside that, the usage of stored procedured in SQL Server will be reused



too without any modification. Examples of stored procedure that have been created in AMS are spDetect, spDetectHoliday, spHalfDayList\_withoutass and etc.

#### ➤ **Confidentiality and Integrity of Data**

Only the authenticated users are allowed to access to the AMS. User's passwords are also encrypted in the database, which has fully enhanced the security of the system. As such, confidentiality and integrity of users' information is greatly protected.

Beside that, before responding to any requests of users from the client-side, SQL server will validate the user account written in "global.asa". For Internet Information Server, "global.asa" is a file with global variable and is protected by web server from remote server. Directory browsing is disabled and the access right of anonymous user that set to log in to the web server is also controlled.

#### ➤ **System Reliability and Accuracy**

AMS is error tolerance whereby it will prompt up the error message if there are any duplications or mistakes made by users. Now, the system can be accessed through internet and is not limited to local area network. However, it does not allow user to clock in, clock out, overtime clock in and overtime clock out beyond the local area network. The system is online 24 hours without any downtime.

In addition to that, all the calculation functions for overtime conversion will be done automatically by system. The accuracy of the overtime balance is guaranteed.

#### ➤ **Simplicity of User Interface**

AMS is designed using frame. The left hand side frame is the options menu while the right hand side frame is the functionality of each control of left hand side frame. All the graphical designs are consistent and quite similar to the GUI of the window environment. Users will find it user friendly, easy to use and understand.



### 7.1.3 System Limitation

#### ➤ Browser and Operating System

AMS is only supported by certain operating system, which are Window 95, window 98 and Window NT with Internet Explorer 4.0 or above. This is because most of the scripts in AMS are written in VB Script, which is not supported by other browser such as Netscape Communicator.

#### ➤ Language Support

AMS is supported by one language, which is English. It is a limitation if anyone of the staffs is come from the foreign country.

## 7.2 Problems Encountered

Various of problems are encountered thorough out the process of AMS development. Some of the problems are critical. However, they are managed to be solved successfully while some of them are not. The following are some of the problem that arises during the development process:

### 7.2.1 Setting Up the Operating System

The setup of operating system (Window NT Server) is critical for the operation of AMS development. Not only the hard disk needs to be formatted, the instructions of installation are complicated. As such, the whole setup process takes a long time. Beside that, lack of experience is also one of the factors in resulting the repeated setup failure.

#### Solution

Reinstall the Window NT Server and follow the instruction sequence properly until the operating system is working. The server needs to be restarted each time after upgrading the service pack and configuring the network setting.



### 7.2.2 Setting Up SQL Mail Services

Even though the SQL Server is installed successfully but the SQL Mail Services cannot be started. An error message is prompted asking for a valid MAPI profile.

#### Solution

SQL Mail must have a mail profile created in the same domain account that is used to start SQL Server. The profile is activated when a user logs to Windows NT successfully. When a mail's stored procedure, such as xp\_sendmail is executed, SQL Mail will look for the defined mail profile in the domain account to send out the fail. So, a mail profile must be created through the mail in control panel. However, Windows Messaging Service or Microsoft Outlook is needed to create a mail profile.

### 7.2.3 Creating activeX dll file to convert datetime format in ASP

Microsoft Interdev does not support the convert format in ASP. It only supports the formatdatetime() function, which is constrained to certain formats only. The date time format like 'dd/mm/yy', 'hh:mm', 'mmmm d, yyyy' can not be used in ASP.

#### Solution

Write a class module in Visual Basic to convert the needed date time format. Then, make it to a dll file. The dll file, which has been created for AMS is 'convertse.dll'. This object can only be used after the dll file is registered to the system server 32.

### 7.2.4 Transferring the Existing Database to the New Database

The first problem arises is some of the tables together with their data from existing database cannot be transferred to the new database. The second problem is the design of the tables from existing database cannot be copied exactly to the new database, which the tables created in new database only consist the name of fields without the key structure.

#### Solution

The first problem is solved by examining the table structure in new database whether it is exactly the same with the existing one. If the table design structure is different, some



changes and modifications need to be made. After the data has been transferred, change again the table design structure according to what it should be.

The second problem is solved by selecting the option “transfer objects and data between SQL server 7.0 database”, not the option “copy table(s) from the source database” before the process of exporting data. By choosing the option, all the design structure of tables will be appeared in new database.

### 7.2.5 Accessing the Session Value from Client Side

The value retrieved from database is stored in session but it can't be accessed from the client side. For example, the user's reminder/event for the particular day is stored in database, when the day comes, it should pop out the reminder once the user login to the system. One of the ways is storing the value retrieved from database in session temporary, then call the session from client side in order to pop out the message.

#### Solution

Using the java script to solve this problem. The VB script cannot perform the scripting in accessing the session value from client side. The example codes are as below:

```
<script language = "javascript">
<!--
    var c
    var Event1 = ' <%=session("Event1")%> ' ;
    var Event2 = ' <%=session("Event2")%> ' ;
    var Event3 = ' <%=session("Event3")%> ' ;

    c = Event1 + '\n'
    if (Event2 != "")
        c = c + Event2 + '\n'
    if (Event3 != "")
        c = c + Event3 + '\n'
    window.alert ("This is your reminder:" + '\n\n' + c)
-->
</script>
```



### 7.3 Recommendations and Future Enhancements

The Backward Overtime Application and Overtime Conversion modules are currently not providing the functions for **revocation**. This may be considered in the future enhancement.

The system may be presented in other **language** other than English. In the aspect of database, it should be able to restore automatically after the disasters like power failure. Any transaction that stopped because of the unexpected broken down of database must be rolled back automatically.

The usage of **smart card** (Touch and Go) can be implemented in this system to provide more secure transaction. Besides, the user does not need to enter any login ID, he or she just needs to show the smart card or 'Touch and Go' for verification. The concept of 'One Card' or "Touch and Go' can apply to other type of application that available in the university. It can replace the Matrix Card that is currently used by the entire student in the university.

### 7.4 Knowledge Gained

The following are the knowledge that have been gained from this project:

#### ➤ A Better Understanding of Network System

In the development of AMS, knowledge in network system must be sufficient. Since AMS is a web system, issue such as shared resources, distributed system, three-tier architecture and network bandwidth are the major concern. As a result, research that has been done on network system gave a better understanding and pictures of how the actual system is running in the network environment.

#### ➤ Ability in Setting Up Platform and Different Kind of Server

The opportunity in setting up the platform, window NT Server 4.0 and other kinds of server such as Microsoft Internet Information Server, Microsoft SQL Server 7.0 and Microsoft Exchange Server 5.5 for AMS has increased my skill and hand-on experience that cannot be obtained from textbooks.



➤ **Learning New Programming Languages**

New Programming Languages such as ASP, VB script and Java script have been learned from this project. Beside that, the skill in writing the good programming has been sharpened out since problems are always discussed and solved together among group members. In addition to that, the knowledge of Visual Basic and Transact SQL has been enriched through daily practices.

➤ **Improving in Database Knowledge**

SQL Server 7.0 is the best database for Microsoft Window Platform. The built-in features of SQL server are powerful such as SQL Mail, Stored Procedures and SQL Server Agent for performing scheduled job. Apart of learning and making use of all these features, knowledge in normalization, table's relationship has been improved too.

## 7.5 Conclusion

As a conclusion, the implementation of Attendance Management System (AMS) has achieved its main objectives as web-based system in automating the entire workflow of attendance logging system of Faculty Computer Science and Information Technology (FCSIT). It has helped in monitoring and managing FCSIT's human capital. The system not only enables user to process attendance transactions but also allowing them to generate timely, accurate and relevant information easily.

In addition, this project has provided the golden opportunity to learn and brush up programming skills especially in web-based programming language like ASP. In line with that, there is an exposure to the ActiveX control, network technology and various of development tools such as Window NT Server, Internet Information Server, SQL Server and Microsoft Exchange Server.

Lastly, this system is said to be successful since all the requirements of the organization have been met. The implementation of AMS in FCSIT has once again proved the feasibility of GOE in an actual office environment.



## References

1. Khai Weyn Ong and Amy Yoon, *The Paperless Office: Myth or Reality*, January 27, 2000  
<http://www.cs.mu.oz.au/~kwo/paperless.html>
2. Phil Neel, Michael Gaheen, Mark Taylor, *Management Issues in a "Paperless" Environment*, Inspection Report No.99-06-01,  
<http://www.sba.gov/IG/paperlessreport.html>
3. McKenna, Joseph F (1995, August). Better than a "Paperless Office", *Management Office Technology Journal*.
4. What is the MSC ?  
<http://www.mdc.com.my/msc/flagship/eg.html>
5. Approach To Realising The Vision Of Electronic Government  
[http://www.mampu.gov.my/EG/EG\\_Approach.htm](http://www.mampu.gov.my/EG/EG_Approach.htm)
6. Generic Office Environment  
[http://www.educom.on.ca/index.html?gen\\_off/contents.htm~basefrm](http://www.educom.on.ca/index.html?gen_off/contents.htm~basefrm)
7. Nex Trak Attendance Management System  
<http://202.233.127.171/nextrakattendancemanagementsy.htm>  
<http://202.233.127.171/Product2.htm>
8. FASTECH Time and Attendance System  
<http://www.f-tech.com/>
9. Viking Attendance Management System  
<http://www.vikingsolutions.com/>



10. The JTA-Jupiter for Windows Time and Attendance System  
[http://www.jantek.com/frames/jta\\_startup.htm](http://www.jantek.com/frames/jta_startup.htm)
11. HawkEye Employee Management Software  
Window based Graphical Attendance, Leave and Overtime System  
<http://www.geocities.com/siliconValley/Heights/9704>
12. State of Maine's Time & Attendance Project (TAMS)  
[http://janus.sate.me.us/bac/BAC/home\\_page/timeandattendance.htm](http://janus.sate.me.us/bac/BAC/home_page/timeandattendance.htm)
13. Itil – Software Solutions for the future about Attendance Management System  
<http://groupware.ital.com>
14. Keogh Software AMS 2000  
<http://keogh-software.ie>
15. "Learning Active Server Page"  
<http://activeserverpage.com>
16. "Microsoft Scripting Technologies"  
<http://www.microsoft.com/vbscript>
17. **Software Engineering** by Mladen A. Vouk, Thu. 08/28/97 FALL 1997: NC State  
CSC510/ECE510  
<http://courses.ncsu.edu/CSC510/Lectures/CSC510.F97.WLS/HTML/lect7/outfile.htm>
18. Ang Tan Fong, *Attendance Management System*, Project Ilmiah Tahap Akhir, 1999/2000.
19. Whitten, Jeffrey L., *System Analysis and Design Methods*, 4<sup>th</sup> Edition, McGraw-Hill



20. Fertuck, Len, *System Analysis & Design*, Wm. C.Brown Communications Inc, 1995
21. Ian Sommerville, *Software Engineering*, 5<sup>th</sup> Edition, Addison-Wesley, 1996
22. Pfleeger, Shari Lawrence, *Software Engineering Theory and Practice*, Prentice Hall, 1998
23. Kenneth E.Kendall, Julie. Kendal, *System Analysis and Design*, 4<sup>th</sup> Edition, 1998
24. *SQL Server 7: the complete reference* / Gayle Coffman
25. Richard Anderson, Chris Blexxrud, Andrea Chiarelli and others, *Professional Active Server Pages 3.0*, Wrox Press Ltd, 1999.
26. Robert Vieira, *Professional SQL Server 7.0 Programming*, Wrox Press Ltd, 1999.
27. Buser, David and Kauffman, John and Others, *Beginning Active Server Page 3.0*, Wrox Press Ltd, 1999.